# User Interface Design

## Lecture 7: Design Principles and Prototyping

Dr. Obead Alhadreti

# Learning Objectives

- Describe user interface elements
- Explain design principles
- Describe prototyping and different types of prototyping techniques

# User Interface elements

- Graphical user interface elements include but are not limited to:

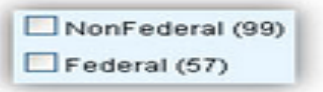
1. **Input Controls**: checkboxes, radio buttons, dropdown lists, list boxes, buttons, text fields, date field.

2. **Navigational Components**: search field, pagination, tags.

3. **Informational Components**: tooltips, progress bar, notifications, message boxes.

4. **Containers**: accordion

# Input Controls

| Element | Description | Examples |
|---|---|---|
| Checkboxes | Checkboxes allow the user to select one or more options from a set. It is usually best to present checkboxes in a vertical list. More than one column is acceptable as well if the list is long enough that it might require scrolling or if comparison of terms might be necessary. |   |
| Radio buttons | Radio buttons are used to allow users to select one item at a time. |   |
| Dropdown lists | Dropdown lists allow users to select one item at a time, similarly to radio buttons, but are more compact allowing you to save space. Consider adding text to the field, such as 'Select one' to help the user recognize the necessary action. |   |

# Input Controls

| | | |
|---|---|---|
| List boxes | List boxes, like checkboxes, allow users to select a multiple items at a time,but are more compact and can support a longer list of options if needed. |  |
| Buttons | A button indicates an action upon touch and is typically labeled using text, an icon, or both. |  |
| Dropdown Button | The dropdown button consists of a button that when clicked displays a drop-down list of mutually exclusive items. |  |
| Toggles | A toggle button allows the user to change a setting between two states. They are most effective when the on/off states are visually distinct. |  <br>  |

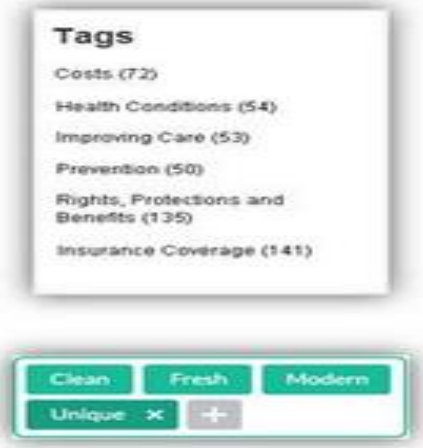
# Input Controls

| | | |
|---|---|---|
| Text fields | Text fields allow users to enter text. It can allow either a single line or multiple lines of text. |   |
| Date and time pickers | A date picker allows users to select a date and/or time. By using the picker, the information is consistently formatted and input into the system. |   |

# Navigational Components

| Element | Description | Examples |
| --- | --- | --- |
| Search Field | A search box allows users to enter a keyword or phrase (query) and submit it to search the index with the intention of getting back the most relevant results. Typically search fields are single-line text boxes and are often accompanied by a search button. |  |
| Breadcrumb | Breadcrumbs allow users to identify their current location within the system by providing a clickable trail of proceeding pages to navigate by. |   |

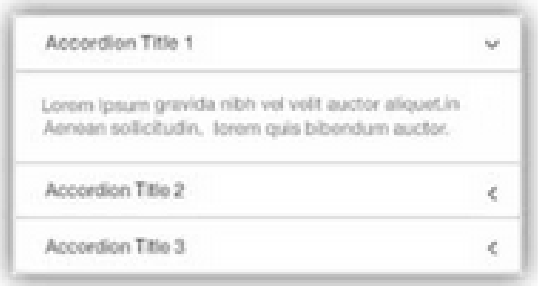
# Navigational Components

| | | |
|---|---|---|
| **Pagination** | Pagination divides content up between pages, and allows users to skip between pages or go in order through the content. |  |
| **Tags** | Tags allow users to find content in the same category. Some tagging systems also allow users to apply their own tags to content by entering them into the system. |  |

# Information Components

| Element | Description | Examples |
| --- | --- | --- |
| Notifications | A notification is an update message that announces something new for the user to see. Notifications are typically used to indicate items such as, the successful completion of a task, or an error or warning message. |   |
| Progress Bars | A progress bar indicates where a user is as they advance through a series of steps in a process. Typically, progress bars are not clickable. |   |
| Tool Tips | A tooltip allows a user to see hints when they hover over an item indicating the name or purpose of the |   |

# Containers

| Element | Description | Examples |
| --- | --- | --- |
| Accordion | An accordion is a vertically stacked list of items that utilizes show/ hide functionality. When a label is clicked, it expands the section showing the content within. There can have one or more items showing at a time and may have default states that reveal one or more sections without the user clicking |  |

# Design Principles

# Donald Norman's principles

- Visibility
  - can I see it?
- Feedback
  - what is it doing now?
- Affordance
  - how do I use it?
- Mapping
  - where am I and where can I go?
- Constraints
  - why can't I do that?
- Consistency

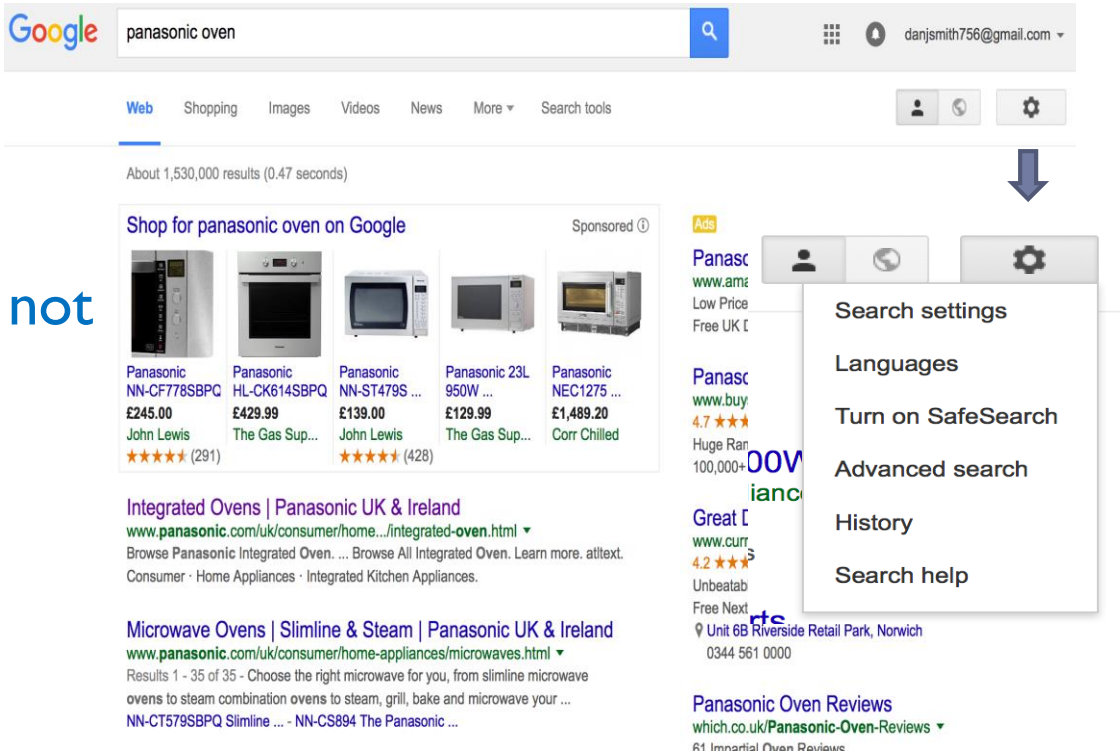
– I think I've seen this before

# Visibility

- The principle of **visibility** suggests that the ease of use of a system is improved when the user can easily see what commands and options are available.
- Controls should be made clearly visible, rather than hidden, and should be placed where users would expect them to be.
- For any complex application, there would be so many buttons that the screen would become crowded and cluttered, and it would be difficult to find the right button. Drop-down lists are an example of a compromise.

# Visibility

Hide functions that are not immediately needed
– e.g. Advanced search



# Feedback

- The system should send information about what is happening back to the user. For example, the system gives users confirmation that an action has been performed successfully (or unsuccessfully).

- Types of feedbacks:

1. Visual is evidence that the control was activated successfully: a button was pressed, a menu option was selected, or a slider was moved to a new position.

# Feedback

2. Auditory

– shutter click in a digital camera

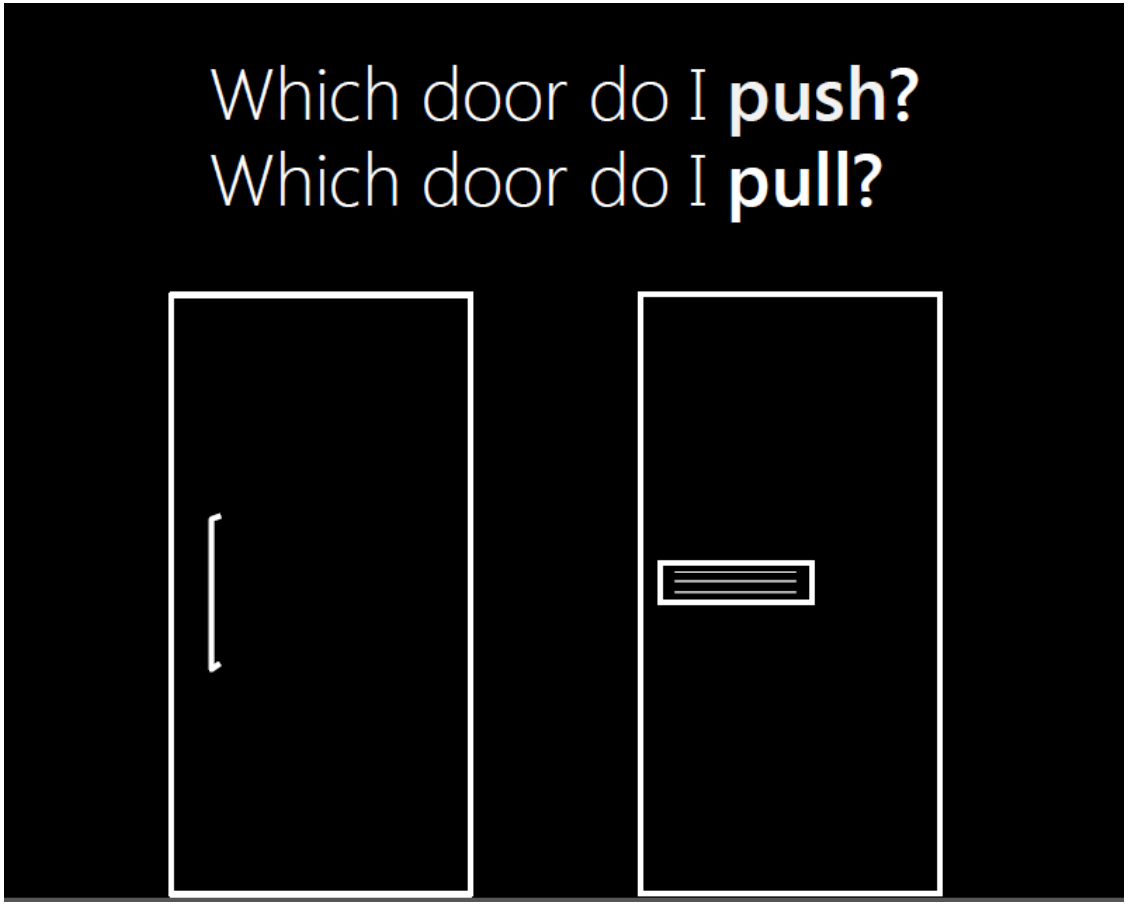
3. Tactile

– movement in keyboard

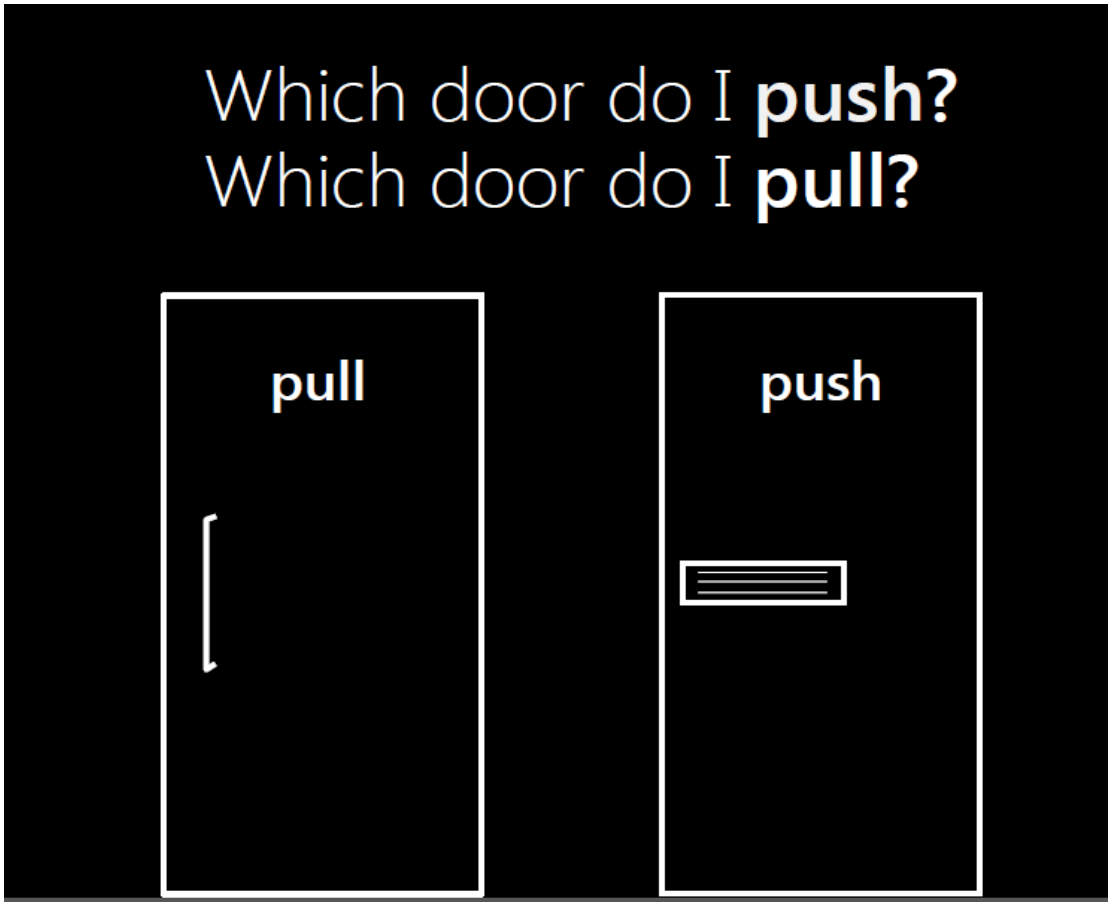
# Affordance

- **Affordance** is a visual attribute of an object or a control that gives the user clues as to how the object or control can be used or operated.

# Affordance



# Affordance



# Affordance

- In user interface design, designers can use different techniques of providing affordance cues:
  - **Pointing**: change the shape of the mouse pointer when the mouse pointer is moved over a control
  - **Tooltips**, or small pop-up messages that appear when the mouse pointer hovers over a control, can provide some additional assistance.
  - **Design conventions** such as hyperlinks

# Mapping

- Pressing a button or activating a control generally triggers the system to perform some function.
- Mapping refers to the relationship between controls and their effect on the computer
- You should always aim to make these mappings as clear and explicit as possible.

# Mapping

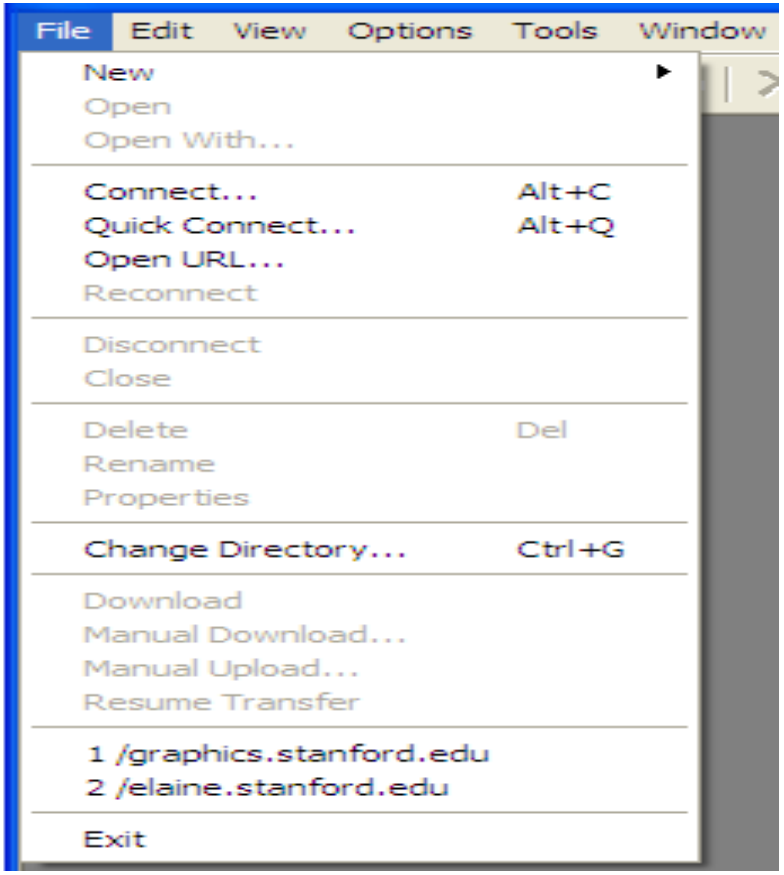
- You can do this by using descriptive labels or icons on buttons and menu items, and by using controls consistently (again, similar controls should have similar behavior and effects).

- Controls should also be positioned in logical ways that match real-world objects or general conventions.

# Constraints

- Restricting interaction to reduce errors
- **Constraints**, or restrictions, prevent invalid data from being entered and prevent invalid actions from being performed.
- For example, word processors disable the "Copy" and "Cut" commands when no text is currently selected.

# Constraints

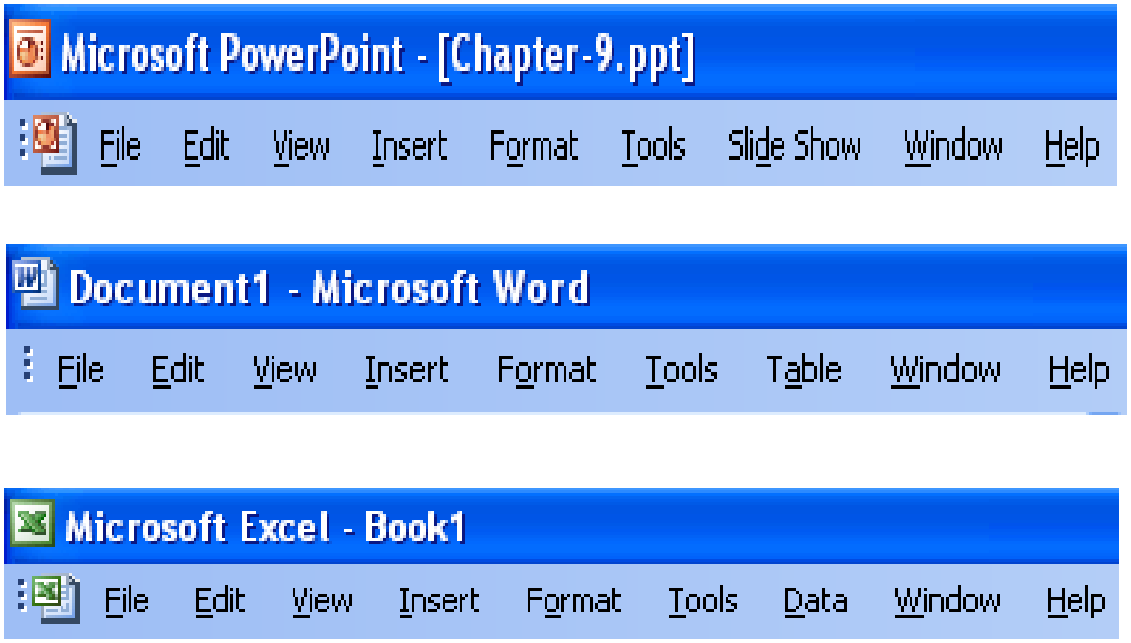


# Consistency

- Similar concepts expressed in similar ways
  - knowledge from previous experiences can be reused
- Similar functions are performed in the same way
- Identical terminology for identical operations
- Inconsistency causes confusion, because things don't work the way the user expects them to.

# Consistency

- Uniformity in appearance, placement, and behaviour



# Other Design Principles

- Nielsen's ten usability heuristics
- Smith and Mosier principles
- Accessibility guidelines

# Prototyping

- **What is a prototype?**

- A draft version of a design before the final version exists
  - "proto" = original or primitive

- A prototype is a simulation of how a finished product will look and work.

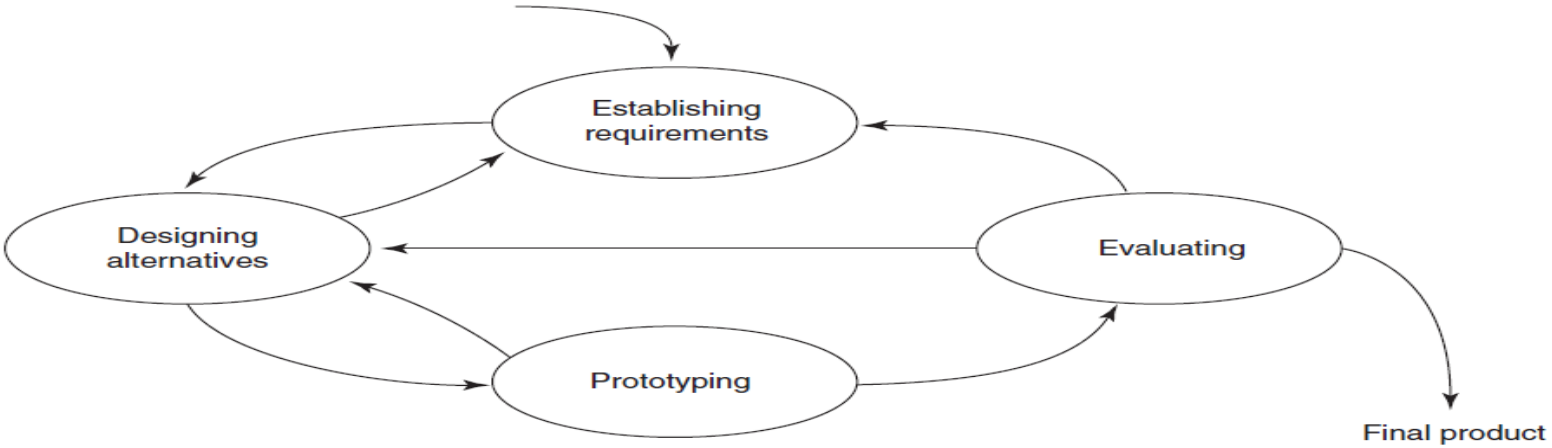
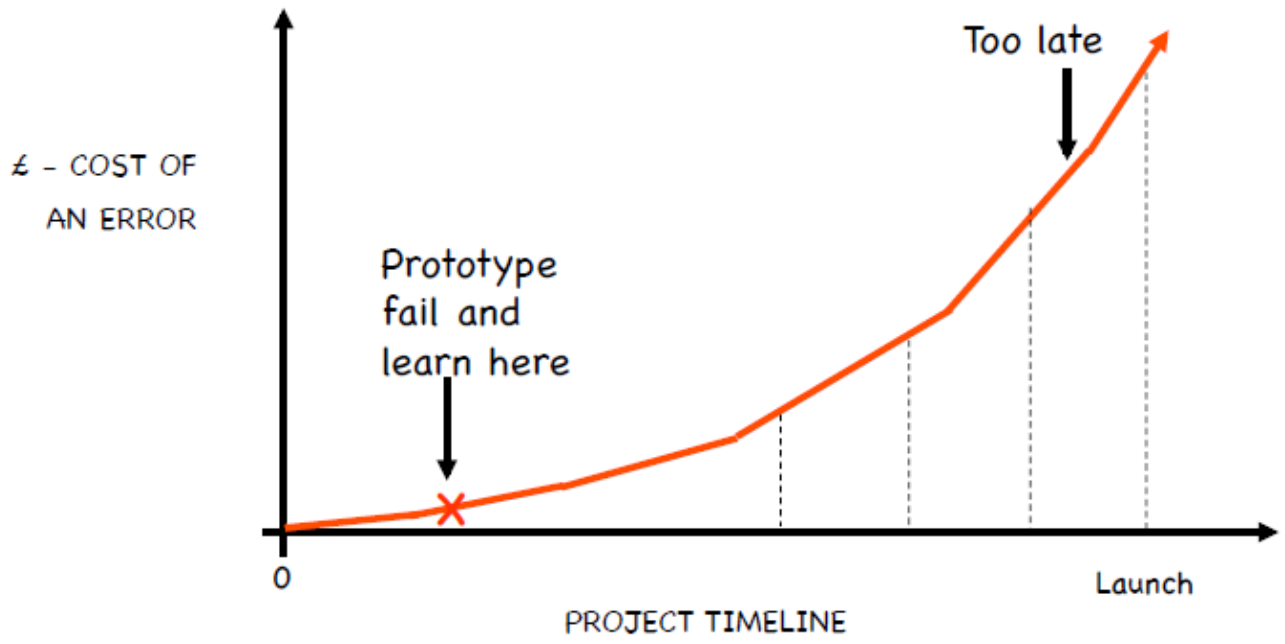



**Figure 9.3** A simple interaction design lifecyle model

# Prototyping

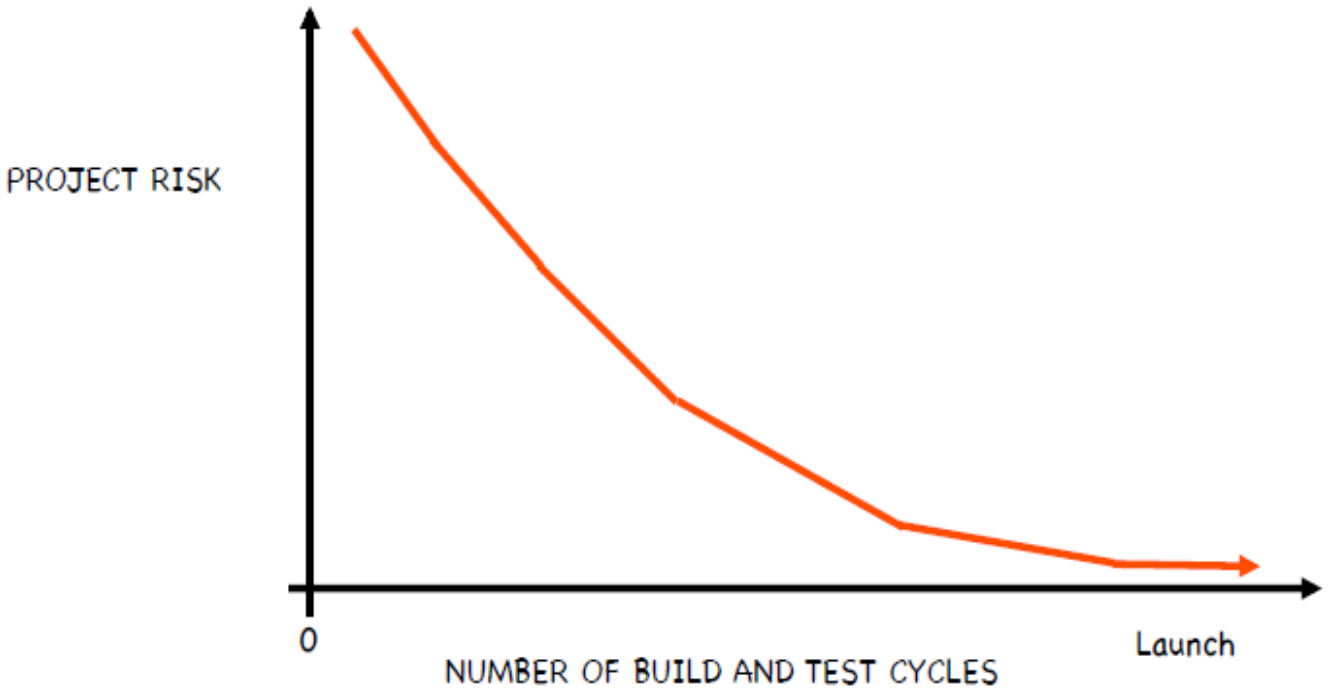
- **Why do we prototype?**
  - Allows stakeholders to interact with an envisioned product, to gain some experience of using it in realistic settings and to explore imagined uses
  - Experiment with alternative designs
  - Facilitate the commination among the development team members (clarifies vague requirements)
  - Get feedback on our design faster (save money)

# Prototyping




* the cost of correcting a problem after a product has been released can be as much as 100 times the cost of resolving it in the development phase

# Prototyping



# Types of Prototyping

1. **Low-fidelity prototypes:**

   has some characteristics of the target product but is otherwise simple and incomplete

2. **High-fidelity prototypes:**

   looks like the final product.

- Product teams choose a prototype's fidelity based on the goals of prototyping, completeness of design, and available resources

# Fidelity in Prototyping

- **Fidelity in prototyping:**

  Fidelity refers to the level of detail.

- Fidelity can vary in the areas of:
  1. Visual design
  2. Content
  3. Interactivity

# Low-fidelity prototyping

- Does not look very much like the final product
- Uses materials that are very different from the intended final version
- Do not allow user interactions
- Used during early stages of development
- Cheap and easy to modify so they support the exploration of alternative designs and ideas
- Used to gather feedback on the basic functionality or visual layout

# Low-fidelity prototyping

- Techniques of low-fidelity prototyping:

1. Paper prototypes
2. Wireframes

# Paper prototypes

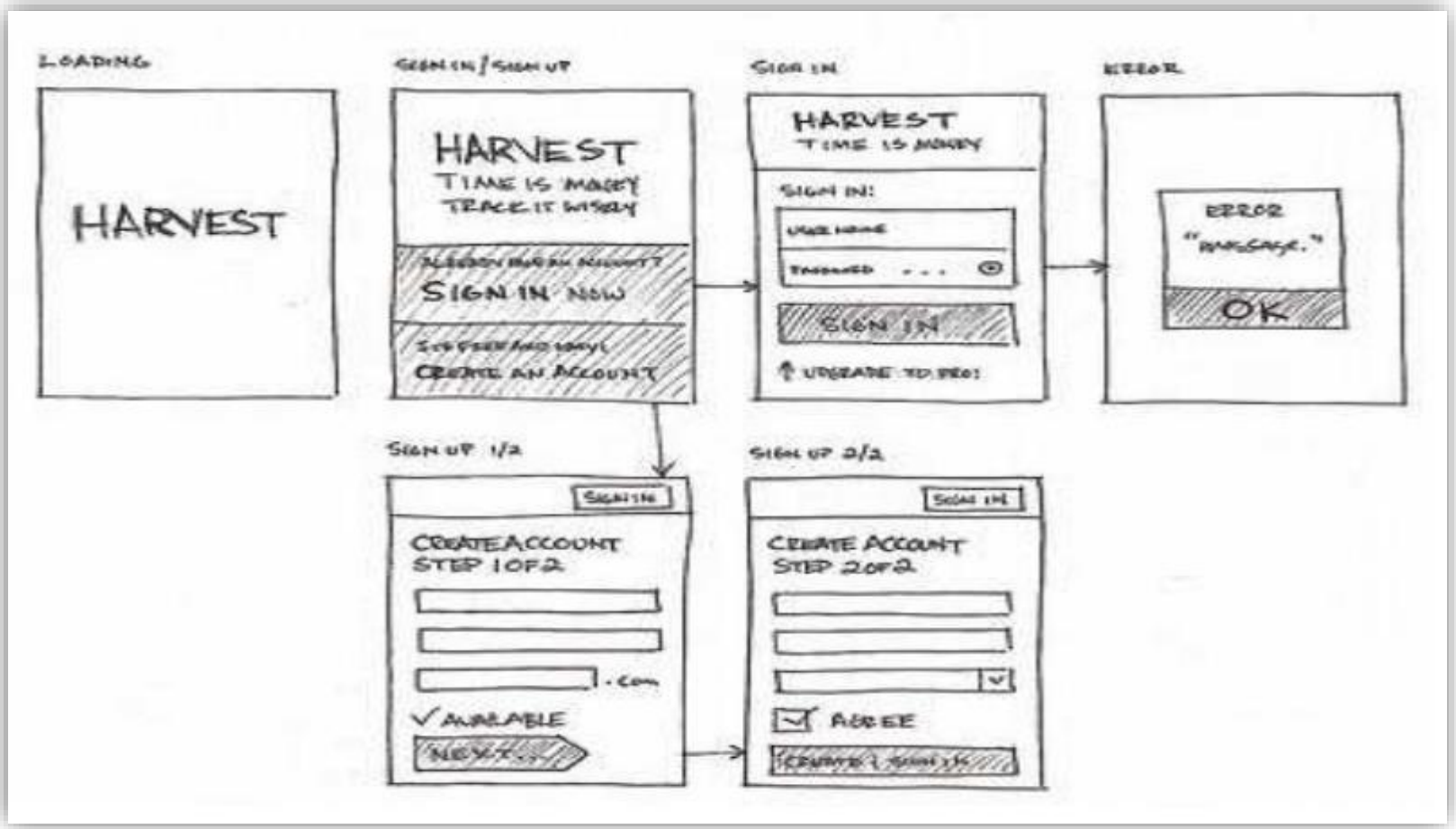
- **Main idea:**
  - Sketch out prototypes of *the interface* on paper
  - Potential users "walk through" task scenarios using the paper interface
  - A designer "plays computer"
  - Other design team members observe & record
  - sketches -> evaluate -> iterate
  - Change the design on-the-fly if helpful - Surprisngly effective

- Widely practiced in industry

# Paper prototypes

- Materials you may need:
  - Large, heavy, white paper (11 x 17)
  - 5x8 in. index cards
  - Post-it notes
  - Tape, stick glue, correction tape
  - Pens & markers (many colors & sizes)
  - Transparencies (including colored)
  - Colorforms (toy stores)
  - Scissors, X-acto knives, etc

# Example 1


LOADING
HARVEST
SIGN IN / SIGN UP
HARVEST
TIME IS MONEY
TRACK IT WISELY
SIGN IN NOW
CREATE AN ACCOUNT
SIGN IN
HARVEST
TIME IS MONEY
SIGN IN:
SIGN IN
ERROR
ERROR
"MESSAGE."
OK
SIGN UP 1/2
SIGN IN
CREATE ACCOUNT
STEP 1 OF 2
.com
AVAILABLE
NEXT
SIGN UP 2/2
SIGN IN
CREATE ACCOUNT
STEP 2 OF 2
AGREE

# Example 2

- **Paper prototyping**

https://www.youtube.com/watch?v=6TbyXq3XHSc

# Wireframes

- Wireframes are simple black and white layouts made using a special computer program that outline the specific size and placement of page elements, site features, conversion areas and navigation for your product.

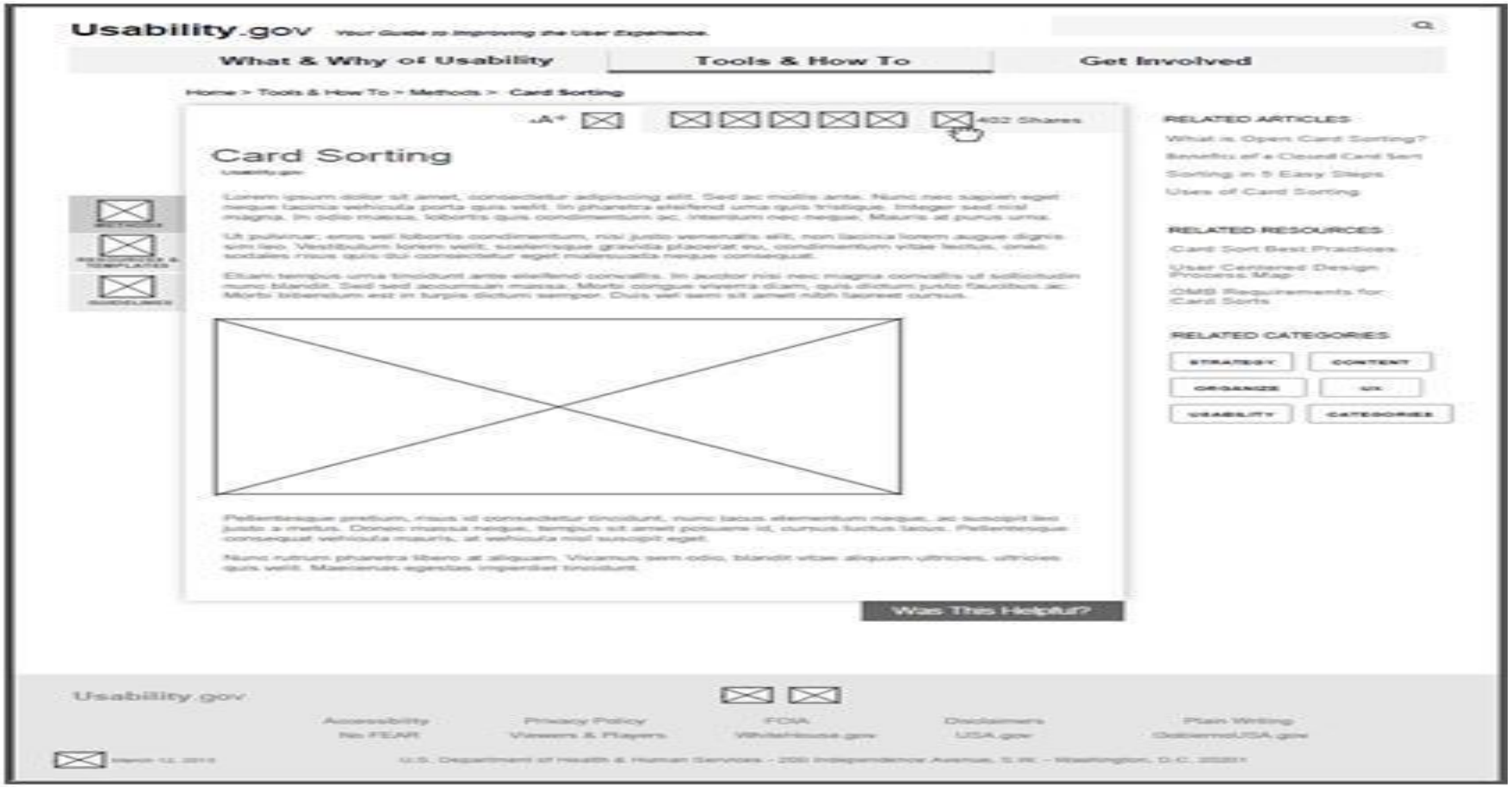
- Wireframes are rough illustrations of page content and structure which serve several purposes throughout the product design and testing process.

- Today, wireframes are most often created using software like Balsamiq or Visio.

# Wireframes

- **When to use wireframing**
  - The idea needs more detail than can be easily captured on paper
  - More layout details,, … need to be explored
  - Clients need to be shown the major design options

# Example


Usability.gov
What & Why of Usability
Tools & How To
Get Involved
Home > Tools & How To > Methods > Card Sorting
Card Sorting
RELATED ARTICLES
What is Open Card Sorting?
Sorting in 5 Easy Steps
Uses of Card Sorting
RELATED RESOURCES
Card Sort Best Practices
User Centered Design Process Map
OMB Requirements for Card Sorts
RELATED CATEGORIES
STRATEGY
CONTENT
ORGANIZE
UX
USABILITY
Was This Helpful?
Usability.gov

# Advantages and Disadvantages of Low-Fidelity Prototypes

| Advantages | Disadvantages |
| --- | --- |
| They are cheap to produce.<br>They can evaluate design ideas and design alternatives.<br>They promote rapid, iterative development. | Their ability to check errors in design is limited.<br>The specification is less detailed so it may be more difficult for programmers to code. |
| They are useful for facilitating communication between users and stakeholders and the UI designer. | A human facilitator is needed to simulate how the UI will work (e.g., by manipulating how different prototypes in response to users actions).<br>Paper may seem less compelling. |
| They can show the look and feel and layout of screens. | They are useful for gathering requirements but are generally thrown away once the requirements have been established. |

# Exercise !

There is a collection of several thousand short video clips and audio recordings. You need to design a system to browse them and allow users to add annotations. Users should be able to retrieve clips based on the name of clips and by the date the clip was made.

Draw a set of lo- prototypes to show the main interactions for this system.

# High-fidelity prototypes

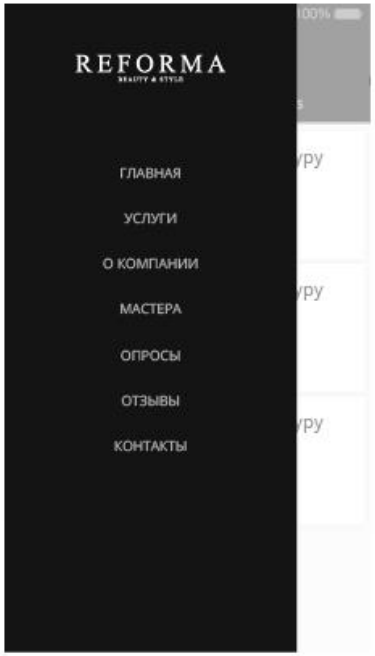
- **High-fidelity prototypes** are computer-based, and usually allow realistic user interactions. High-fidelity prototypes take you as close as possible to a true representation of the user interface.

- Techniques of high-fidelity prototyping:

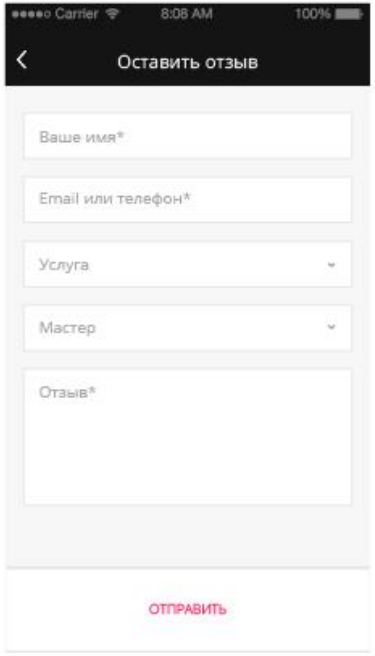
1. HTML/CSS Mockups
2. Clickthrough Prototypes

# Mockup

- A **mockup** is a high fidelity, static, full color design representation. It demonstrates, in a static form, the visual side of the project according to the structure of information and the basic functionalities.





# Clickthrough Prototypes

- **Clickthrough prototypes** is already very close to the finished product. Here, processes can be simulated and user interaction can be tested.

- A prototype looks very similar to the finished product. Early prototyping can save a lot of development costs and time so that the work of back-end product architecture will not be in vain because of unreasonable user interface design.

- A prototype is an excellent tool to obtain user feedback and to test the product.

# High-fidelity prototypes

| Prototype | Advantages | Disadvantages |
| --- | --- | --- |
| High-fidelity prototype | - Interactive<br>- look and feel of final product<br>- clearly defines navigational scheme | - more expensive to develop<br>- time consuming to build<br>- developers are reluctant to change something they have crafted for hours |