

Building User Interfaces

# Usability Evaluation: Rapid Methods

Professor Bilge Mutlu

# Announcements

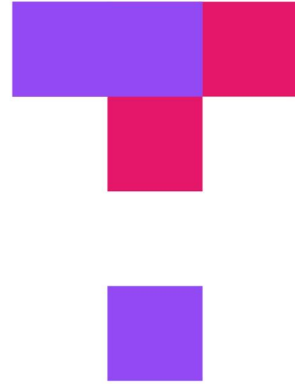
>> Supplemental material

Check class  
schedule  
page!

# What we will learn today?

- >> What is usability evaluation?
- >> What are rapid/expert methods?
  - >> Heuristic evaluation
  - >> Cognitive walkthrough
- >> Assignment preview

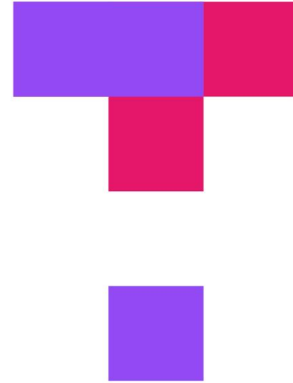
# TopHat Attendance



**TOP HAT**



# TopHat Questions



**TOP HAT**

# What is usability evaluation?

# Usability Evaluation

**Definition:** The assessment of the effectiveness of and user satisfaction with design solutions.

how well does the design work for the user?  
how much does the user like the design?

# Types of Usability Evaluation

1. **Testing-based** methods
2. **Expert-review-based** methods

have users test the design  
(can be time consuming)

design team or experts perform  
(usually fast)

# Testing-based methods

**Definition:** Empirical, i.e., based on data, testing with users who represent the target population of design solutions.

We will cover testing-based methods later in the semester.

# Expert-review-based methods

**Definition:** Also called "usability inspection," review-based evaluation by experts who follow well-established protocols to inspect the usability of design solutions.

Our focus today will be on expert-review-based methods.

# Which methods?

*very*

*common*

*not as  
widely used*

**Heuristic evaluation**, heuristic estimation, **cognitive walkthrough**, pluralistic walkthrough, feature inspection, consistency inspection, standards inspection, formal usability inspection.

We will cover the most commonly use two methods.

# Heuristic Evaluation



# Heuristic Evaluation

**Definition:** Developed by Jacob Nielsen, heuristic evaluation involves having a **small set of evaluators** examine the interface and judge its compliance with recognized usability principles (the "heuristics").<sup>1 2</sup>

usually of members of the design team established principles

---

<sup>1</sup> NN/g: [How to conduct a heuristic evaluation](#)

<sup>2</sup> NN/g: [Video explanations of the 10 heuristics](#)

**NN/g**

**Heuristic  
Evaluation**



# 1: Visibility of system status<sup>3</sup> <sup>4</sup>

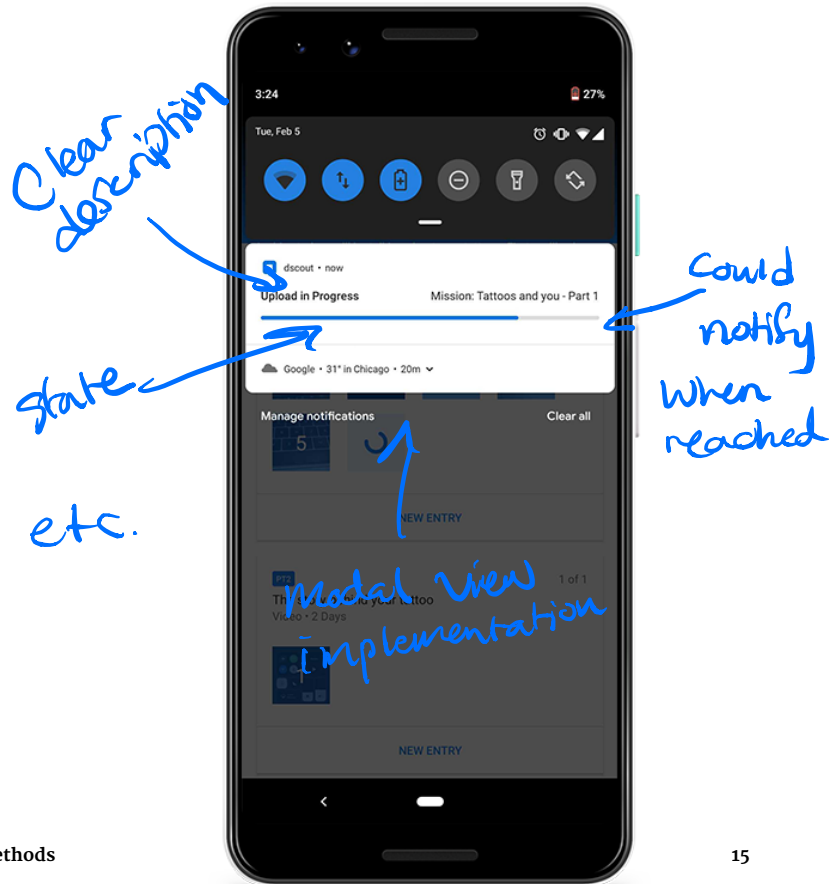
The system should always keep users informed about what is going on, through appropriate feedback within reasonable time.

E.g., email clients making a swoosh sound when sending email.

Think ← loading bars, etc.

<sup>3</sup> NN/g: Visibility of system status

<sup>4</sup> Image source



## 2: Match between system and the real world<sup>5 6</sup>

*The system should speak the users' language, with words, phrases and concepts familiar to the user, rather than system-oriented terms. Follow real-world conventions, making information appear in a natural and logical order.*

*Can be through metaphor*

<sup>5</sup> NN/g: Match between system and the real world

<sup>6</sup> Image source



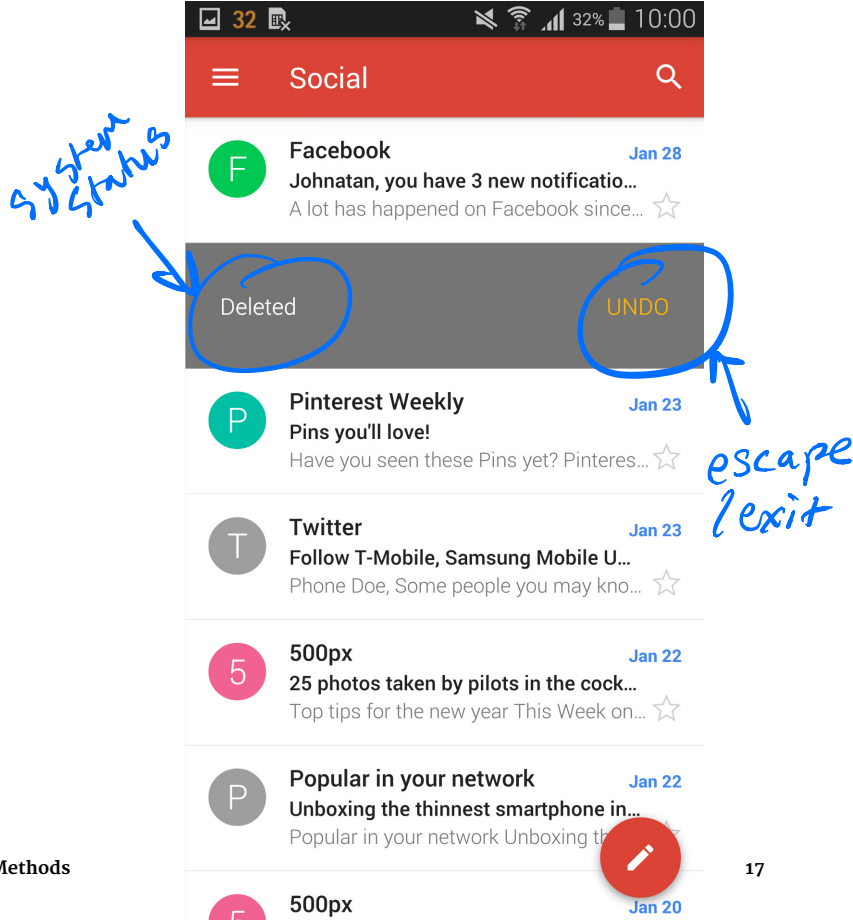
### 3: User control and freedom<sup>7 8</sup>

*Users often choose system functions by mistake and will need a clearly marked "emergency exit" to leave the unwanted state without having to go through an extended dialogue. Support undo and redo.*

E.g., undo for delete/archive in email clients

<sup>7</sup> NN/g: User control and freedom

<sup>8</sup> image source



## 4: Consistency and standards<sup>9 10</sup>

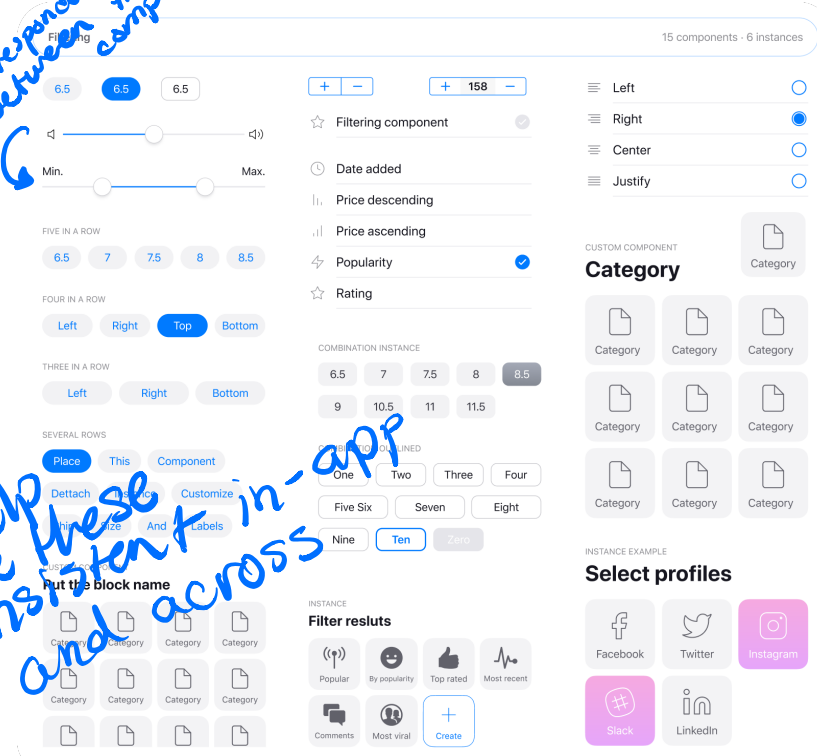
Users should not have to wonder whether different words, situations, or actions mean the same thing. Follow platform conventions.

E.g., component libraries to achieve consistency within an app; platform conventions to achieve consistency across apps.

<sup>9</sup> NN/g: Consistency and standards

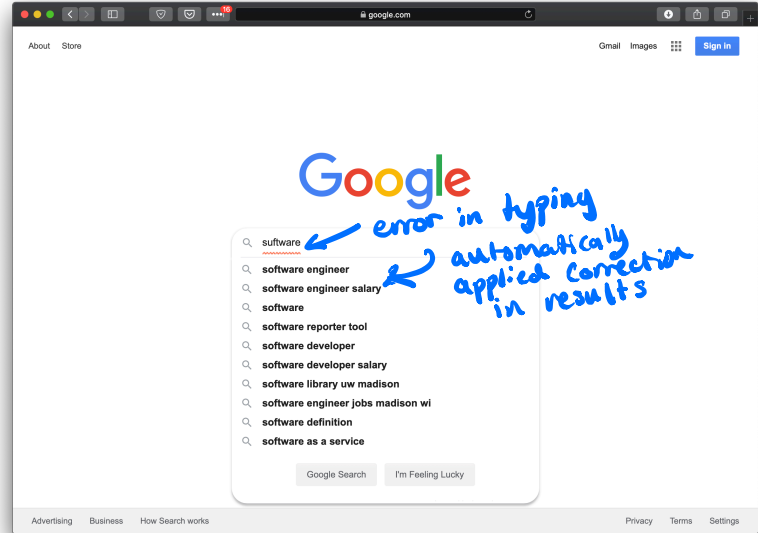
<sup>10</sup> Image source

*see correspondence between these components*



## 5: Error prevention<sup>11</sup>

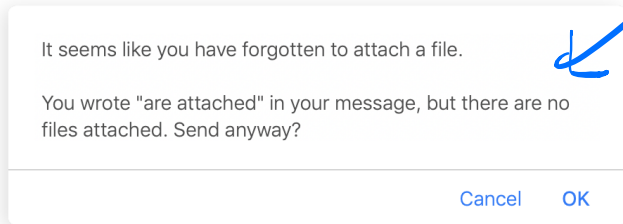
*Even better than good error messages is a careful design which prevents a problem from occurring in the first place. Either eliminate error-prone conditions or check for them and present users with a confirmation option before they commit to the action.*



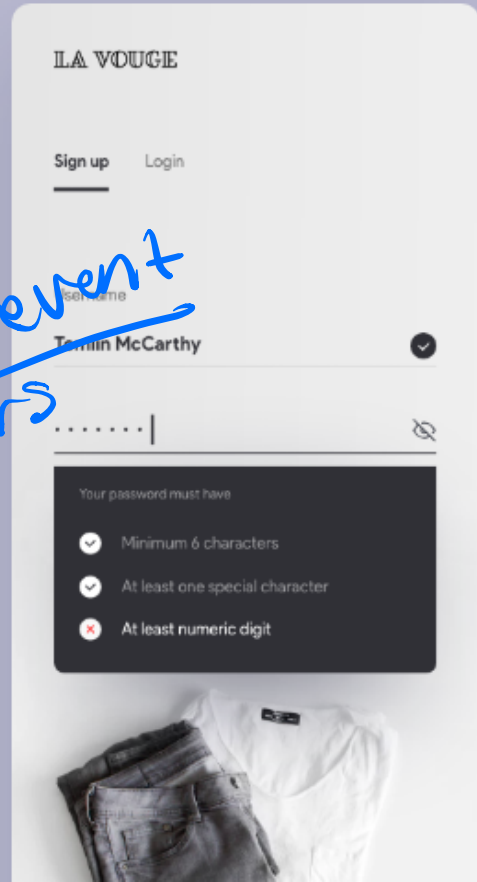
<sup>11</sup> NN/g: Error prevention

## Examples:<sup>12</sup>

- » Autocorrect in search
- » Real-time feedback on new user names, password strength, etc.
- » Attachment reminders in email clients



help prevent errors

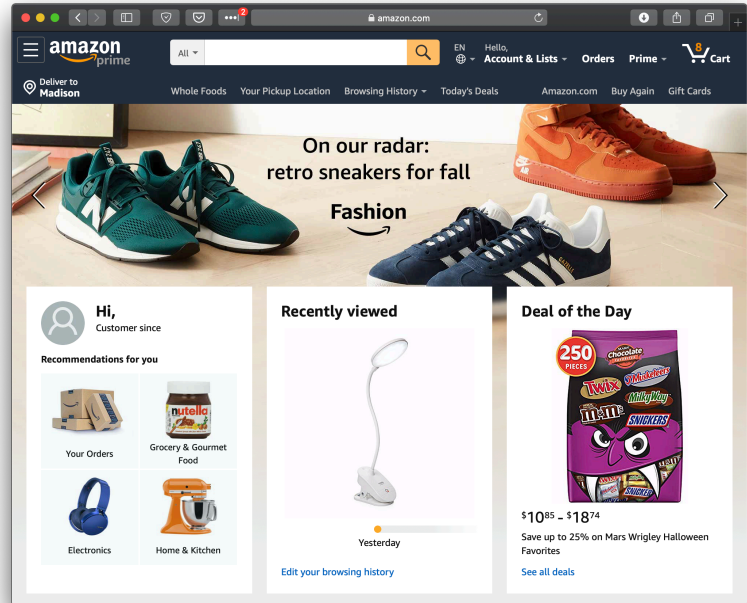


<sup>12</sup>Image source: [Left](#), [Right](#)



## 6: Recognition rather than recall<sup>13</sup>

*Minimize the user's memory load by making objects, actions, and options visible. The user should not have to remember information from one part of the dialogue to another. Instructions for use of the system should be visible or easily retrievable whenever appropriate.*



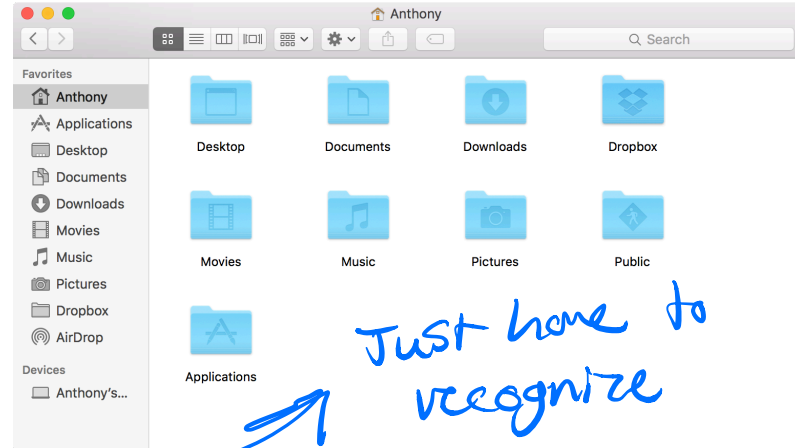
<sup>13</sup> NN/g: Recognition rather than recall

```

bryan — -bash — 80x24
Last login: Tue Nov 28 13:08:25 on ttys001
5K-Stack:~ bryan$ sudo su
>Password:
[sh-3.2# passwd
Changing password for root.
New password:
Retype new password:
[sh-3.2# exit
exit
5K-Stack:~ bryan$ █

```

*require recall*  
*"how do I see all the contents of this folder?"*



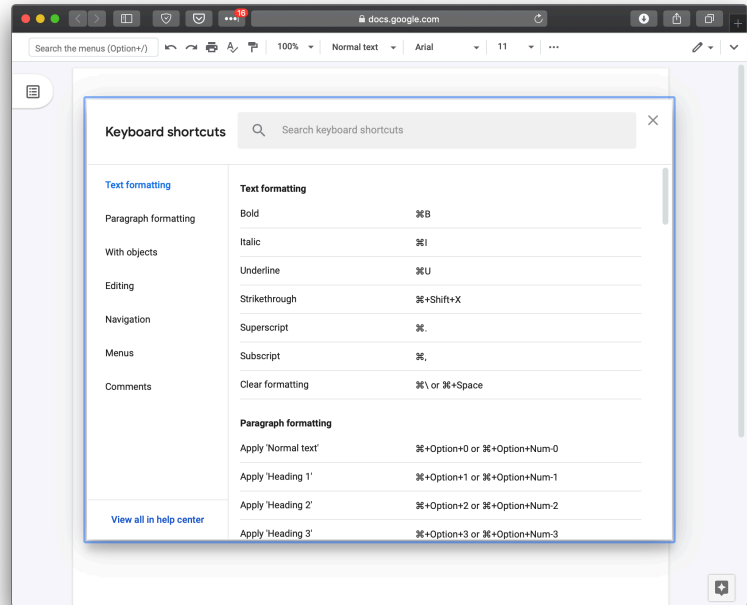
<sup>14</sup> Image source: [Left](#), [Right](#)

## 7: Flexibility and efficiency of use<sup>15</sup>

Accelerators — unseen by the novice user — may often speed up the interaction for the expert user such that the system can cater to both inexperienced and experienced users. Allow users to tailor frequent actions.

keyboards shortcuts

"CTRL + Z"

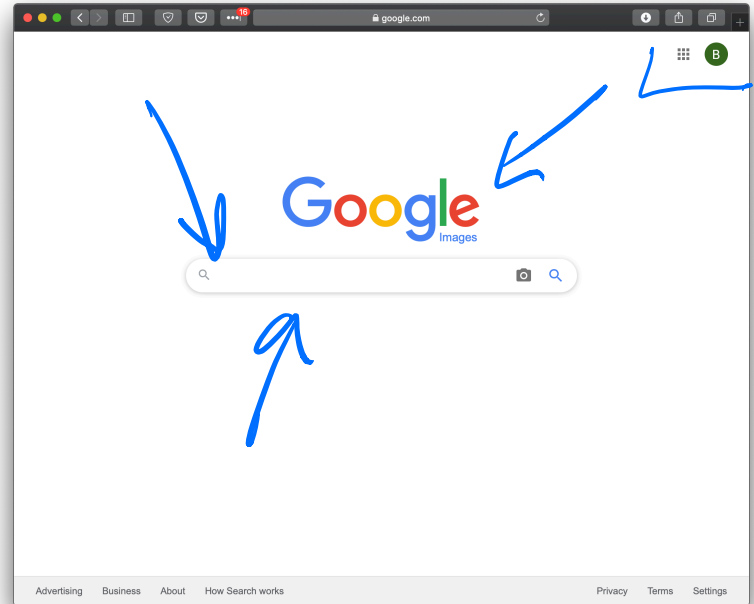


<sup>15</sup> NN/g: Flexibility and efficiency of use

## 8: Aesthetic and minimalist design<sup>16</sup>

*Dialogues should not contain information which is irrelevant or rarely needed. Every extra unit of information in a dialogue competes with the relevant units of information and diminishes their relative visibility.*

*Extra reading:  
Edward Tufte*



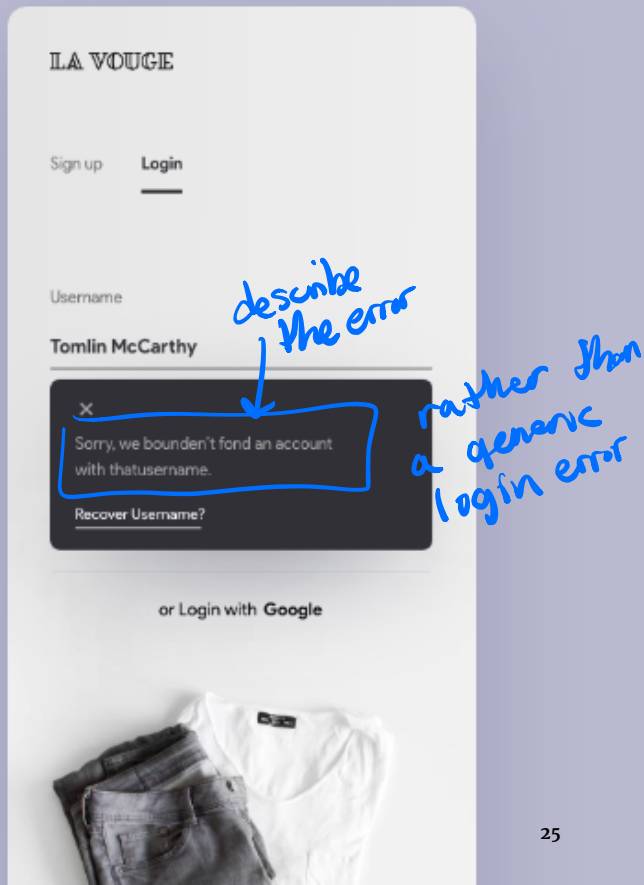
<sup>16</sup> NN/g: Aesthetic and minimalist design

## 9: Help users recognize, diagnose, and recover from errors<sup>17 18</sup>

Error messages should be expressed in plain language (no codes), precisely indicate the problem, and constructively suggest a solution.

<sup>17</sup> NN/g: [Help users recognize, diagnose, and recover from errors](#)

<sup>18</sup> [Image source](#)



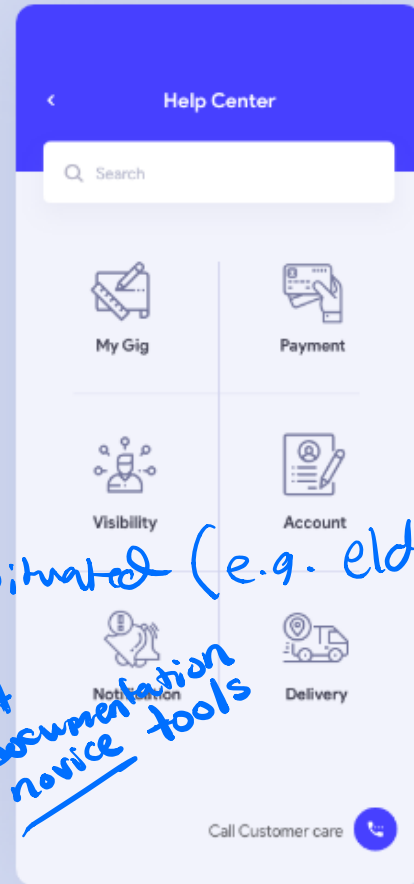
## 10: Help and documentation<sup>19 20</sup>

Even though it is better if the system can be used without documentation, it may be necessary to provide help and documentation. Any such information should be easy to search, focused on the user's task, list concrete steps to be carried out, and not be too large.

6 Best

- ① when you can help certain users get situated (e.g. elderly)
- ② expert tools

generally:  
should not  
need documentation  
for novice tools



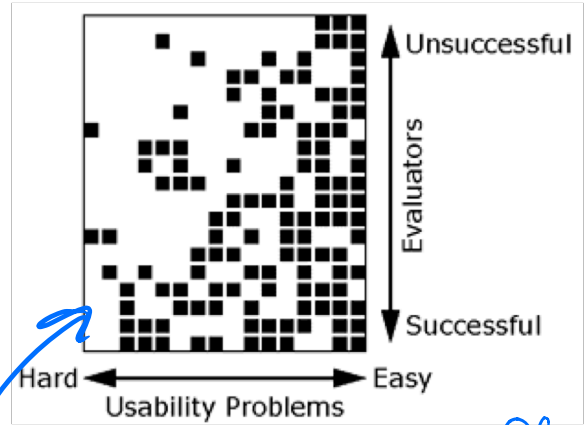
<sup>19</sup> NN/g: Help and documentation

<sup>20</sup> Image source

# Process<sup>21</sup>

1. Identify 3-5 usability experts with domain knowledge; determine the heuristics to use;
2. Each inspector individually reviews as feature/screen/page for each heuristic;
3. Inspectors merge and prioritize their findings, brainstorm solutions, report conclusions.

*could be members of the team or outside experts*



*different raters find different issues, especially when rarer or hard*

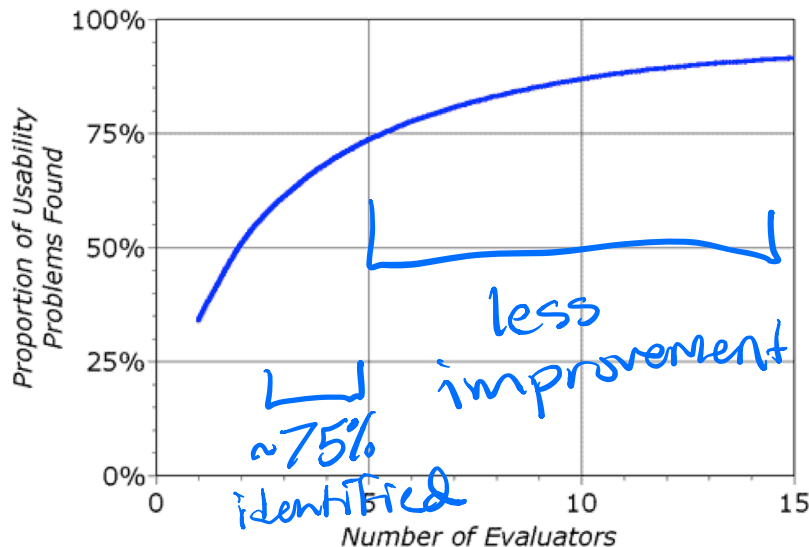
<sup>21</sup> NN/g: [How to Conduct a Heuristic Evaluation](#)

## How many evaluators are needed?

The rule of thumb is 3-5.<sup>22</sup>

$$ProblemsFound(i) = N(1 - (1 - l)^i)$$

- »  $i$  independent evaluators
- »  $N$  total number of usability problems in the interface
- »  $l$  the proportion of all usability problems found by a single evaluator



<sup>22</sup>[Image source](#)



# Heuristic Evaluation Reporting<sup>23</sup>

**Definition:** A document that highlights the top three to five usability problems and suggested solutions.

good example



---

<sup>23</sup> [A good heuristics evaluation report example](#)

Heuristic evaluation reports usually include:

1. Prototype screen, page, location of the problem
2. Name of heuristic
3. Reason for reporting as negative or positive
4. Scope of problem
5. Severity of problem (high/medium/low)
6. Justification of severity rating
7. Suggestions to fix
8. Possible trade-offs (why the fix might not work)

] context

# Severity Ratings

Code	Category	Recommendation
4	Usability catastrophe	<i>Imperative to fix before product can be released</i>
3	Major usability problem	<i>Important to fix, so should be given high priority</i>
2	Minor usability problem	<i>Fixing this should be given low priority</i>
1	Cosmetic problem only	<i>Need not be fixed unless extra time is available on project</i>
0	~	<i>Team does not agree that issue impacts system usability</i>

*More severe* ↑

# Pros & Cons of Heuristic Evaluation

*"discount"  
method*

## Pros:

- >> Inexpensive and intuitive
- >> Can be used frequently and any time during the design process
- >> Effective at early stages of design
- >> Serves as a training tool for designers

## Cons:

- » Does not capture all aspects of usability
- » Does not provide a comprehensive understanding of the interaction
- » Might discourage user testing
- » May result in false positives

sometimes need to add your own

doesn't really handle the dynamics of interaction

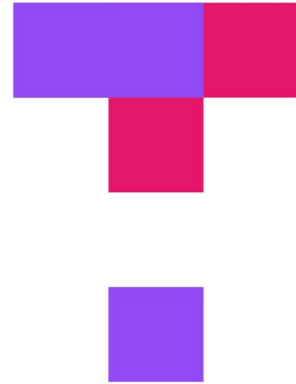
sometimes find the wrong solution to the problem

still necessary!

# Further Reading on Heuristic Evaluation

- >> [UX Collective article with great examples](#)
- >> [Videos and articles by Jacob Nielsen](#)

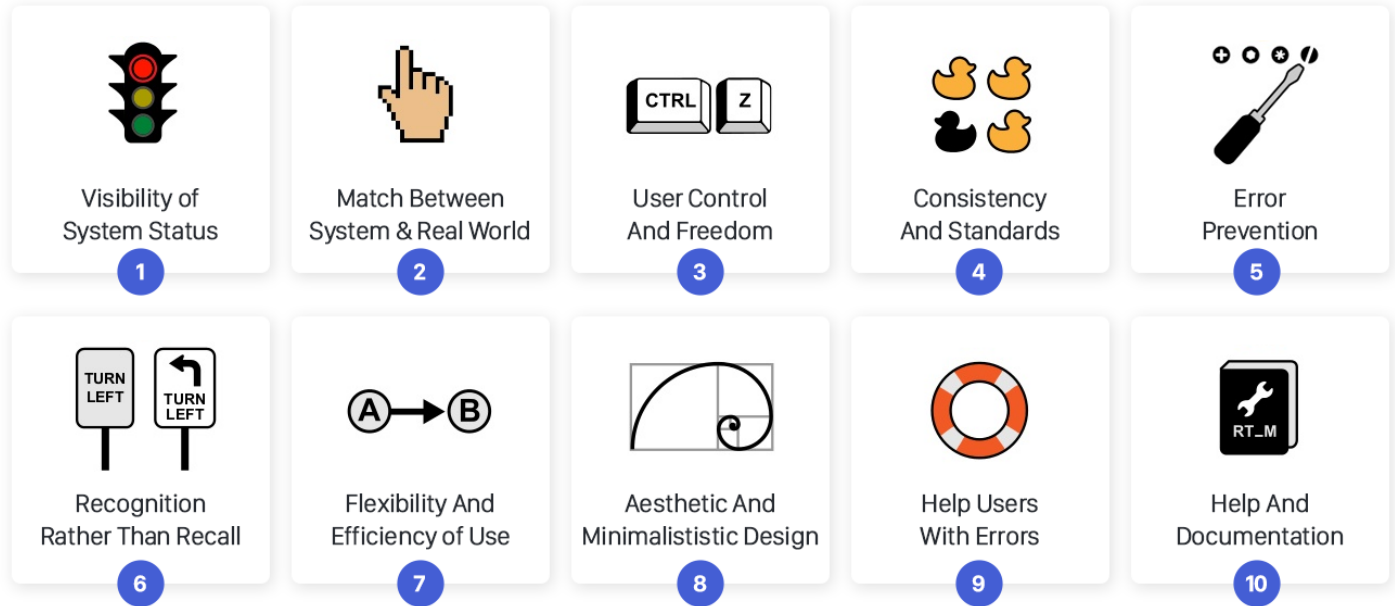
# TopHat Quiz



**TOP HAT**

# In-Class Activity





# Cognitive Walkthrough

# Cognitive Walkthrough

**Definition:** Expert review method where a usability specialist assesses the learnability and discoverability of a design by posing and answering a set of questions.

how well will  
a new / novice  
user navigate through  
the design?

# What do we need to perform a cognitive walkthrough?

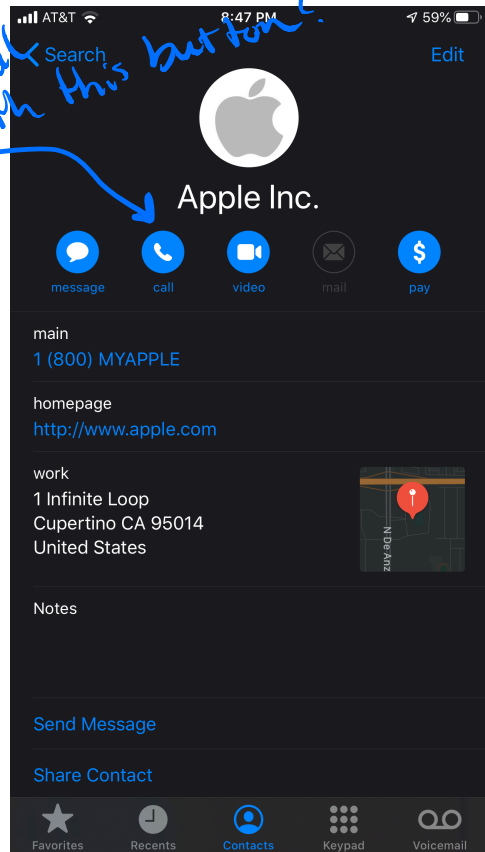
- » A prototype ← the design
- » A user profile ← the archetype user considered "imagined"
- » Set of tasks ← what they are going to do
- » Sequences of actions ← steps to perform each task

# Question 1<sup>25</sup>

Will the user try and achieve the right outcome?



*Will they try to call with this button?*

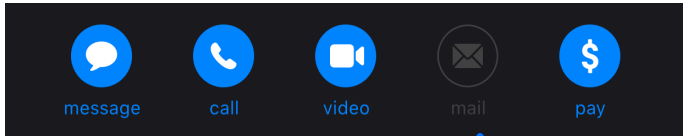


<sup>25</sup> Image source

## Question 2

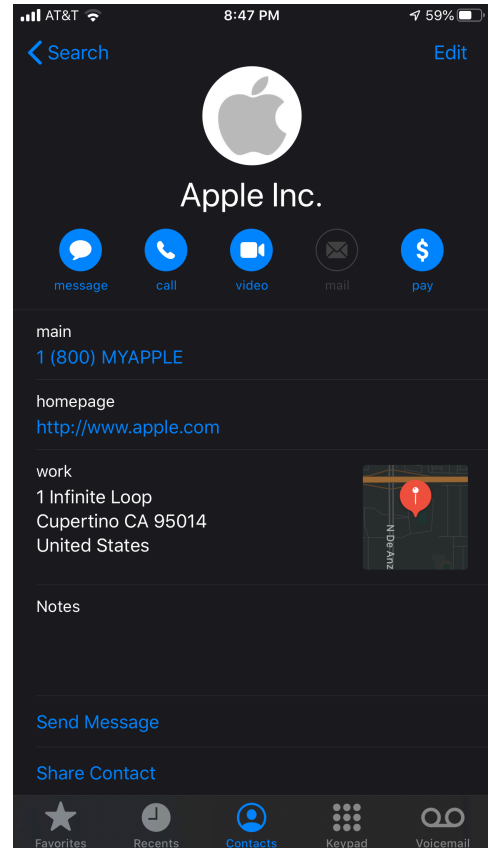
*Will the user notice that the correct action is available to them?*

Does the design support the user's mental model, knowledge, and prior experience to achieve the outcome?



can call

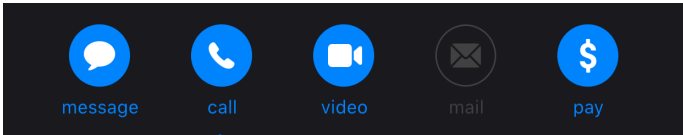
can't email



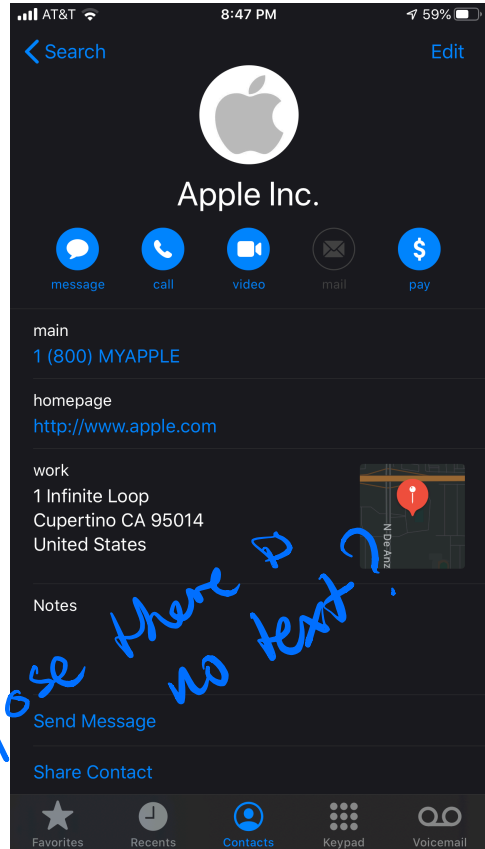
# Question 3

*Will the user associate the correct action with the outcome they expect to achieve?*

Does the visibility, availability, and accessibility of the design element support the action?

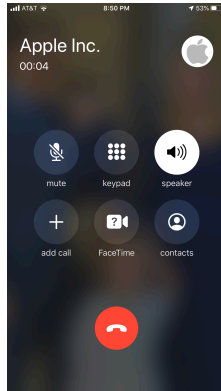


*↑ will they associate this with calling? suppose there is no text?*

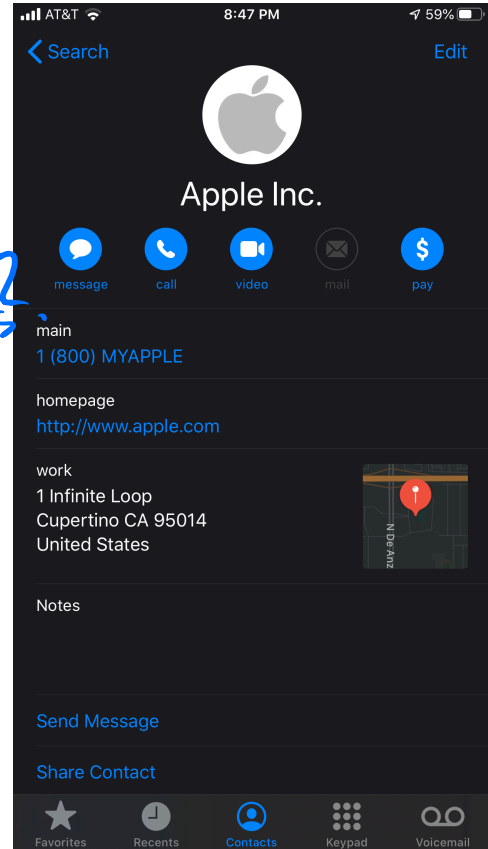


# Question 4

*If the correct action is performed; will the user see that progress is being made towards their intended outcome?*




*will they see this?*





# Pros & Cons of Cognitive Walkthrough

## Pros:

- >> Powerful for:
  - >> Walk-up-and-use interfaces  *Kiosks, ATM's, etc*
  - >> New concepts/forms of interaction
  - >> Systems designed for various user profiles
- >> Can be performed frequently and at any stage of the design process

# Pros & Cons of Cognitive Walkthrough

## Const:

- >> Focuses only on discoverability/learnability ← narrow
- >> Best when used with usability testing

# Further Reading on Cognitive Walkthrough

- » [UX Collective: Assessing usability with Cognitive Walkthrough](#)
- » [Usability body of knowledge](#)
- » [Cognitive Walkthroughs](#)
  - » [Cognitive walkthrough template](#)

# In-Class Activity

# Assignment Preview

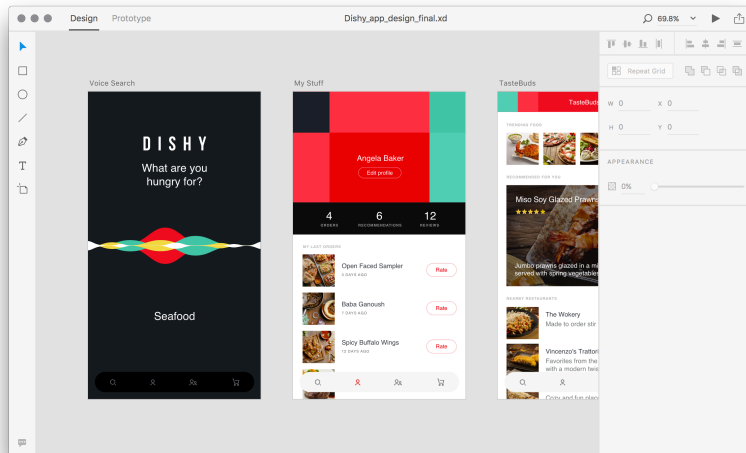
# Design 06: Usability Heuristics (2 points)

You will practice heuristic evaluation to improve your design for the Module 1 deliverable.

- » **Step 1.** Prototype your design
- » **Step 2.** Review the heuristics
- » **Step 3.** Identify potential violations
- » **Step 4.** Develop design recommendations
- » **Step 5.** Update your design

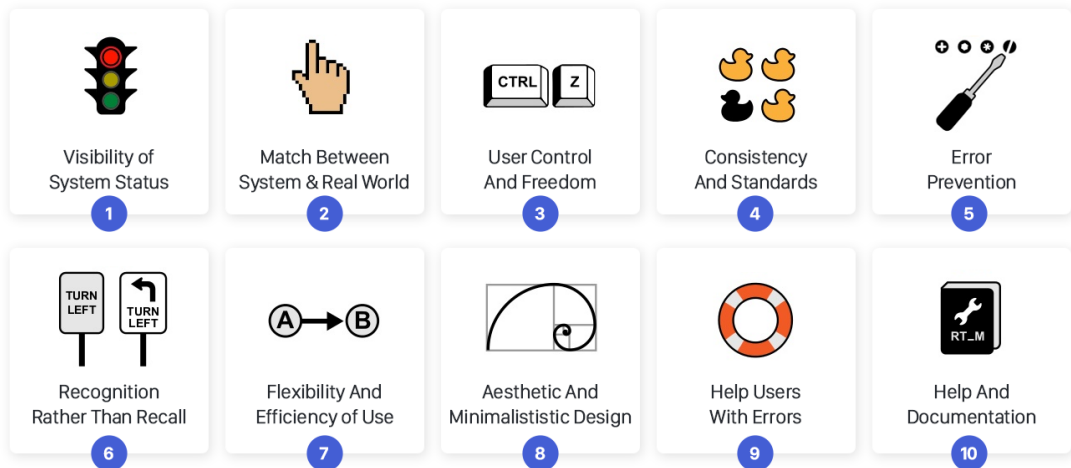
## Step 1 Prototype your design<sup>26</sup>

Create a working prototype using Adobe XD that captures the main functionality of your design.



<sup>26</sup> [Image source](#)

## Step 2. Review the heuristics<sup>27</sup>



<sup>27</sup> [UX Collective](#)



## Step 3. Identify potential violations<sup>28</sup>

Report the violations you identified in a table and mark them on screenshots.

Determine severity.

Usually, heuristic evaluation sheets are used for this step. We will use a simpler table.

Heuristic Evaluation Sheet

Device	<input type="radio"/> 0 I don't agree that this is a usability problem at all
Evaluator	<input type="radio"/> 1 Cosmetic problem only; need not be fixed unless extra time is available on project
Date	<input type="radio"/> 2 Minor usability problem; fixing this should be given low priority
Website/App	<input type="radio"/> 3 Major usability problem; important to fix, so should be given high priority
Browser/OS	<input type="radio"/> 4 Usability catastrophe; imperative to fix this before product can be released
Task/Feature	

1. Visibility of system status	Issues	Recommendation
<p>The system should always keep users informed about what is going on, through appropriate feedback within reasonable time.</p> <p>Severity</p> <p><input type="radio"/> 0 <input type="radio"/> 1 <input type="radio"/> 2 <input type="radio"/> 3 <input type="radio"/> 4</p>		
<p>2. Match between system and the real world</p> <p>The system should speak the users' language, with words, phrases and concepts familiar to the user, rather than system-oriented terms. Follow real-world conventions, making information appear in a natural and logical order.</p> <p>Severity</p> <p><input type="radio"/> 0 <input type="radio"/> 1 <input type="radio"/> 2 <input type="radio"/> 3 <input type="radio"/> 4</p>		

Credit by Hsin-Jou Lin

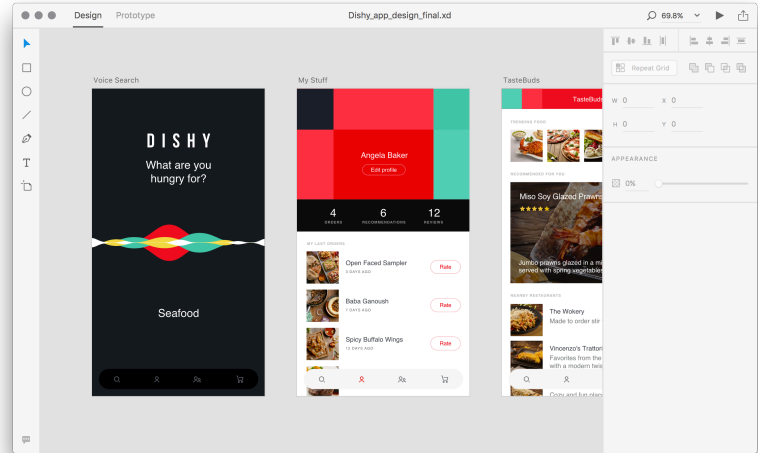
<sup>28</sup> [Image source](#)

## Step 4. Develop design recommendations

For each violation, recommend a design solution.

## Step 5. Update your design<sup>29</sup>

An updated Adobe XD prototype that will serve as the basis for your Module 1 deliverable.



<sup>29</sup> [Image source](#)

Good luck!

# What did we learn today?

- >> What is usability evaluation?
- >> What are rapid/expert methods?
  - >> Heuristic evaluation
  - >> Cognitive walkthrough
- >> Assignment preview