Hello class, and welcome to our new voice-over-slides video lesson set.

The last two sets addressed ways in which we could evaluate interfaces with users, some of the psychologic effects connected with them, and ethical considerations when utilizing them.

In this set, we will cover a very different approach to evaluating interfaces, making use of fundamental design principles and heuristics as the framework for an expert review of a system.

We will start with the overall concept and process, and then move on to some specific examples of each heuristic.

This video set will be longer and more detailed and sets the stage for much of Phase 3 as well as the next individual homework assignment. You might want to pause at times to take a short break.

**Design Principles and Usability Heuristics**

Heuristic Evaluations: An “introspective” method that can be used to inspect an interface for usability problems.

The heuristics are also good to keep in mind when trying to avoid common design pitfalls...
**Design principles and usability heuristics**

The design principles represent a broad set of general rules based on research and experience that also describe features of “usable” systems.

- broad usability statements that guide a developer’s design efforts
- derived by evaluating common design problems across many systems

**Heuristic evaluation:** Take these same principles and use them to “evaluate” a system for usability problems.

- Reasonably popular approach since actual user involvement is not required (cheaper and logistically easier) and it end up catching many design flaws.
- Is considered an “expert review” technique.

We have seen many elements of good design principles this semester. The list that I’ll be presenting soon is an attempt to describe high-level principles that can be met using things we have discussed.

This list is based on research and industry experiences. It is not the only such list, but it is the one we will use and it is a commonly used one.

Also, they way we will be using the list is to help us not only think about what flaws an interface might have, but also to work to categorize them to help explain to the designers why the things we identify as being potentiality problematic to users are being seen that way.

In the end, the exact list we use won’t impact the overall process much, as long as we are using one of several good lists. Think of tricks to see what a person wrote on the previous sheet of paper, pencil rubbing, dark powder, multiple photos with lights at different angles, electrostatic detection, they are all good techniques and get you useful results if you use them right.
The expert review in general, and specifically the heuristic evaluation approach, can be seen as advantageous since they can be faster and less expensive than user studies. It reveals a prototype to fewer people, and also can result in a more simplified report to the developers more easily.

The usefulness of the results depends greatly on the abilities of the evaluators to be able to see through the eyes of the users and in practice is often followed up by user studies on a revised prototype.
Details on the “discount” angle

Relative to user-observational studies, this can be cheap and fast and relatively easy for trained practitioners, which can be critical in today’s product cycle...

- There are no special labs or equipment needed.
  - For many things, likely able to run it on your own machine in your office
  - Interesting bonus: can even be used on paper prototypes

- Doing this type of evaluation can be done on the order of one day where other usability testing could take weeks.

- Once the approach is understood by a team it can be used in many scenarios with little additional learning and the more careful you are, the better the results get.

With a good evaluation team, the approach we are looking at can be done in a day or two, and can be more easily and efficiently done with lower-fidelity prototypes than user studies could be.

Also, once an evaluation team has a process in place, it’s not a high-intensity day or two either, so several can be done in quick succession and reasonable pricing is likely.
**Design principles and usability heuristics challenges**

### Challenges (for lack of a better word)

- These principles can’t be treated as a simple checklist.  
  Note: “If done wrong, that’s bad” is a common “disadvantage”, but it is worth noting here.

- There are subtleties involved in their use and in classifying some specific issues that are raised.

- Some consider this a stage before “real” user testing to catch many issues before users are brought in for usability/performance testing.

One of the things anyone using this approach, whether the evaluators or the product team, needs to avoid is thinking of the list of principles as a checklist. It’s not that you “do” each thing on the list, it’s that every aspect of an interface needs to conform to the applicable principles.

It’s also not always an easy decision of which principle to use to describe a problem, so it will be important to inform the product team how to make use of the classifications in the report.

In practice, while a team might have some disagreement on which principle to use to classify a specific issue, in terms of intensity it will be closer to a discussion of whether a tomato is a vegetable or a fruit rather than a discussion of what the best operating system is.
This idea of using design principles as the framework for an expert review was originated by Jakob Nielsen in the early 90s.

He seems to have started with the “Eight Golden Rules” given by Ben Shneiderman in the mid-80s and then refined and iterated on that. He has advocated for different lists over the years.

In brief, an evaluation team tends to have three or four members who each review the interface with usability in mind, and then they meet to share and aggregate their findings.

As previously noted, a variety of design principle lists can be utilized in this technique. Some of them have a very different feel to them. We will use one of Nielsen’s but if you’re curious, you can search online for other lists, such as the list of cognitive engineering principles by Jill Gerhardt-Powals.
To go into a bit more high-level detail, as the members of the evaluation team review the interface, they will tend to walk through it several times, thinking about actual tasks and keeping all of the design principles in mind. They will then combine their individual findings into a team report.

When they present the product team with the report, the product team will hopefully go through the list of design principle “violations” to rework their interface to fix the identified problems.
Thinking about the work cycle of the evaluation team, there are four phases.

First, the team might need to learn about the domain in which the interface will be used and should review representative task scenarios supplied by the design team. An evaluation firm might have different teams that work with different categories of clients to streamline this phase.

Second, the teams go off to do their individual evaluations and then come back together to create their aggregated list. There are multiple ways to aggregate lists.

Once the aggregated list of problems has been made, each needs to be assigned a severity rating. Each member might suggest the rating they would give it, and then negotiate to a final team rating.

The final phase is to present the report to the design team. This will usually include an overview of how to read the report and going through a few example higher-priority suggested fixes.
A side-note on the first phase is that if the system is meant to be used by “anyone” there might not be a need for training or full scenarios, but usually even if “anyone” might use it, there will still be some context to understand.

Getting down into the details, each evaluator might do a walk-through of the system to get an overview of the flow and scope of possible tasks and identify things to explore in more detail. They can then do a second, more detailed pass and make their detailed list of problems, and possibly organize them thematically.

One way to aggregate the lists of problems is to use a sort of Boggle game approach and have one person read their whole list and see whether anyone else has each item, and then merge them.

Alternatively, each member of the team can state one, and then do the merge discussion. Then after everybody has had a chance to state one, they each have the opportunity to state their next, etc, etc. In this second approach, you might go through them in descending order of perceived severity.
The list of design principles on this page are the ones we will be using as our usability heuristics. Please pause to review them.

Welcome back. The next video will explore examples of each of these.
When it comes to the severity ratings, the evaluation team should assume the ratings assigned will be used by the product team to decide which problems to address and how much effort to put into addressing them.

Additionally, it might help a development team realize they need more in-house focus on usability.

When deciding on the rating, you need to consider how often the problem will come up, like whether it’s a one-time sort of thing like setting up a new printer or a recurring one like printing each thing correctly.

Also, when it is encountered, how big of an impact will it have on the user when it does, and will the user learn from the experience or will it be a persistent frustration.

Each person on the evaluation team notes their severity rating opinion, and then this is also assessed and aggregated by the team.
This slide presents Nielsen’s suggested rating values and meanings. Please pause to review them.

Welcome back. Please realize that different domains will naturally have different definitions of things like “usability catastrophe” as well as the rest.

Someone told me that at NASA a usability catastrophe is one that leads to the loss of the craft, so clearly teams will likely want to define what the ratings mean to them.
## Examples of individual entries

<table>
<thead>
<tr>
<th>Can’t copy info from one window to another.</th>
</tr>
</thead>
<tbody>
<tr>
<td>violates “Minimize the users’ memory load”</td>
</tr>
<tr>
<td>severity: (3) major</td>
</tr>
<tr>
<td>fix: allow copying</td>
</tr>
<tr>
<td><strong>Typography uses mix of upper/lower case formats and fonts.</strong></td>
</tr>
<tr>
<td>violates “Consistency and standards”</td>
</tr>
<tr>
<td>slows users down</td>
</tr>
<tr>
<td>probably wouldn’t be found by standard user testing</td>
</tr>
<tr>
<td>severity: (1) cosmetic</td>
</tr>
<tr>
<td>fix: pick a single format for entire interface</td>
</tr>
<tr>
<td><strong>Green flashing lights mean system settings are being changed, red lights mean normal functionality taking place.</strong></td>
</tr>
<tr>
<td>violates “Consistency and standards”</td>
</tr>
<tr>
<td>could confuse new users</td>
</tr>
<tr>
<td>severity: (2) minor issue, cosmetic</td>
</tr>
<tr>
<td>fix: reverse color usage of lights</td>
</tr>
</tbody>
</table>

This slide presents some examples of how individual heuristic entries might appear. Please pause to review them.

Welcome back.
For the debriefing, this is when the evaluation team shares a summary of their report and touches on some representative highlights.

Realize that the development team might be naturally defensive if only on a subconscious level.

Focus on the idea of *critique not criticize* and having a conversation where the evaluation team perhaps gets a feeling of how resource-intense some of the suggested changes will be for the team, but also conveys why the usability issues being fixed will improve the customer experience.
Why Multiple Evaluators?

Single “average” evaluator achieves lower-quality results…

Previous comparisons indicated that an average single evaluator will only find around 35% of the actual usability problems but that using five evaluators will lead to finding around 75% of them (some don’t get found until the users try it out). Granted, if you have top-notch evaluators, this can be higher, like 90%.

It is good to understand that the expert review team is a team for multiple reasons.

Multiple evaluators are more likely to find more usability issues but are also in a position to help sculpt the descriptions of the issues, the reasons why they are issues, as well as the suggestions for fixes.

It’s also important to accept the limitations and that even with a good team, due to the time frame and fact that they are not actual users, the likelihood is that up to a quarter of the issues might not be found - this is actually based on experiments done in the 90s.
Adding evaluators beyond three or four adds to the complexity of the process and to the cost without strong cost/benefit tradeoff to justify it.

In the next two videos of the set we will explore the nine design principles we’ll be using as our heuristics in much more detail.