CMPT 363: User Interface Design
Summer 2021

Week 3: Design Heuristics, Usability Testing

Instructor: Victor Cheung, PhD

School of Computing Science, Simon Fraser University

Recap from Last Lecture

- User Interface Design
  - What are user interfaces
  - Why do we need to design them
- HCI & UX
- Importance of good user interfaces
- How do we design good user interfaces
  - Usability goals
  - Design principles

Today

- Evaluating Interfaces
- Heuristic Evaluation
- Usability Testing (pre-recorded)

- Assignment 1 is available on Canvas (based on heuristic evaluation), ask questions at Canvas Discussion
- Project Part 1 will be available on Canvas by the end of this week, you can start forming teams though
- Office hours
  - Gayatri Ganapathy gganapat@sfu.ca | Mondays 6p-7p @Zoom (link can be found in the Teaching Team page on Canvas)
  - Xinyi Zhong xza248@sfu.ca | TBA

Assignment 1

- Good/bad/interesting UI examples
  - Illustrate with photos & annotation

"the lights on the buttons provide feedback to user when ready/selected"
OR
"the sign provides no clear instructions to user on how to do X (they might have done Y without knowing its wrong)"

NOT:
buttons have lights, or instructions are unclear

- Heuristic Evaluation
  - Isolate problematic/good elements

"Labels used in the interface come from terms in the classroom, and presented with familiar icons at consistent locations"

NOT:
Corresponding heuristics are "matching between system & real-world" and "recognition rather than recall"

Forming Teams

- Project of 5 (max.)
  - Use the Discussion to coordinate (or the CSSS Discord?)
  - Recommended to form 5 of your own. Those with less than 5 might be assigned with someone without group
- Use Canvas Collaborations to assign yourselves into pre-defined groups (https://community.canvaslms.com/docs/DOC-10516-421264913)
  - Set a regular meeting time (at least once per week), know any time conflicts
  - Decide on a collaboration platform (e.g., Google Docs, Microsoft Teams, Dropbox, …etc)
  - Agree on and sign the Team Contract together
- Project description will be available on Canvas by the end of this week

Evaluating Interfaces

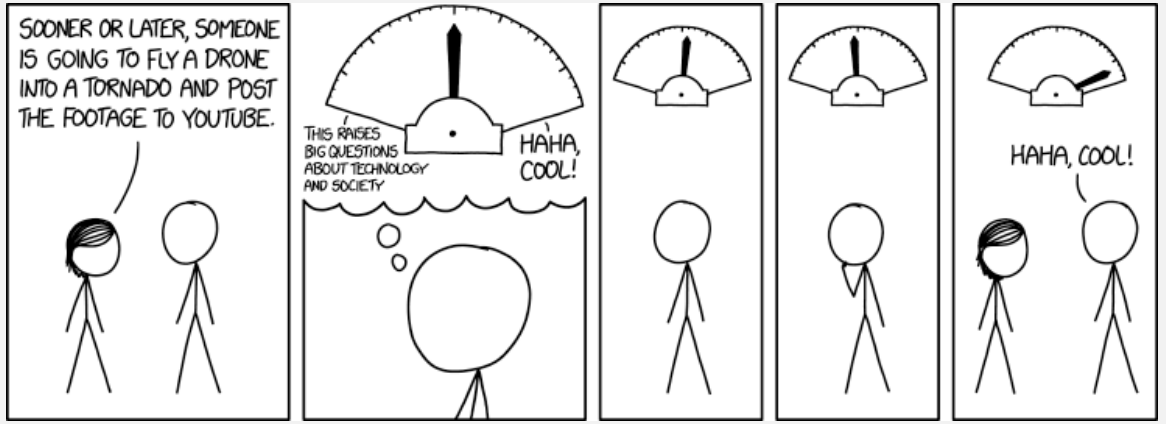
Why? What? Where? When?

Why Do We Want to Evaluate a User Interface?

- Chat activity – think of one reason
  - write it in the chat window, but don't press enter yet

Why Do We Want to Evaluate a User Interface?

- Users expect more than just a usable system
  - Not just "does it get the job done?", but also "how well does it get the job done?"
- Can be conducted before product launch to discover flaws (and fix them)



https://xkcd.com/2072/

What to Evaluate?

- All levels
  - From prototype to final product
- All parts
  - From individual components to its entity
- All attributes
  - Aesthetics, safety, speed, error rate, duration

Where to Evaluate?

- Laboratory
  - More control on the procedures & environment, good to pin-point causal relationships (internal validity)
  - For example, invite people to a facility and measure how fast they can find something in the interface
- Natural setting (aka in-the-wild)
  - More realistic scenarios, typically more generalizable to the rest of the public (external validity)
  - For example, give people an evaluation unit and have them try it out at home for a week

- Depends on what to evaluate
  - For example, performance vs experience, specific vs general

When to Evaluate?

- Any time!
  - The idea is to make sure users' needs are being met at every step during the design process
  - It can be a formative process where the interface is evaluated while it is being formed (qualitative-driven), or a summative process where the outcome of using the interfaces is evaluated (quantitative-driven)

"when the cook tastes the soup, that's formative;
when the guests taste the soup, that's summative." – R. Stake. 2004. p17

Stake R. Standards-Based and Responsive Evaluation. Sage Publications 2004.

Issues about Evaluation

- Factors that could impact validity of evaluation
  - The surroundings (e.g., weather, noise, distraction)
  - Bias (e.g., users doing the evaluation, questions being asked)
  - Hawthorn effect (people modify their behaviour due to awareness of being observed)
- Ways to mitigate these factors
  - Better control of the environment
  - More users doing the evaluation
  - Better design of the evaluation process (will talk more about this in Week 12 – experimental design)

Activity

- MC Question – Which of the following is *not* an expected benefit from usability testing?
  - Validate the design with end users
  - Make potential customers want to buy your product
  - Identify issues of the interface
  - Develop empathy towards end users

Types of Evaluation (ID-Book Ch 14.3)

- Controlled settings directly involving users
  - Usually done in labs to provide the most control (mostly called usability testing/studies/experiments)
- Natural settings involving users
  - Usually done outside labs where the interface is designed to be used at (mostly called in-the-wild studies)

- Any settings not directly involving users
  - Consultants/field experts instead of users (mostly called analytical evaluation)

Analytical Evaluation

- Done without involving users, instead by experts to predict user behaviour and to identify usability problems based on knowledge of usability, users' behaviour, the contexts in which the system will be used, and the kinds of activities that users undertake (ID-Book, 14.3.3)
- Useful for uncovering key design issues before usability testings are conducted
- Examples:
  - Heuristic Evaluation (we'll cover this today)
  - Cognitive Walkthrough (details in later lectures)
  - Fitt's Law Analysis (details in later lectures)

Activity & Break (5+5min)

- Think of a frustrating moment you had when using a software (e.g., unexpected behaviour, got stuck)
  - Talk about what caused the frustration and if you figured out how to handle it
  - And what would you have designed it otherwise

- Do it for approx. 5 minutes, followed by a break of 5 minutes
  - Share with class on the chat when back

Heuristic Evaluation

Part of Discount Usability Engineering

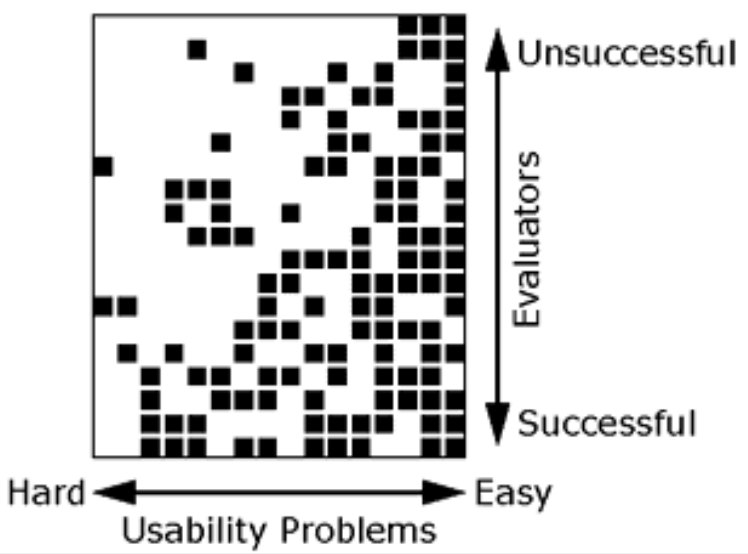
Discount Usability Engineering

- "Something is better than nothing" – Jakob Nielsen
  - Simple user testing (5 users)
  - Simple prototypes (for early testing)
  - Heuristic evaluation

- Quick read (and a 3m16s video) on Discount Usability: 20 Years, by Jakob Nielsen https://www.nngroup.com/articles/discount-usability-20-years/

The Heuristic Evaluation Method

- Experts evaluate the user interface using guidelines to "inspect" its usability
  - Involving multiple evaluators can cover more usability problems



Key findings

- *the easier problems are found by many evaluators*
- *the harder problems are found by only a few evaluators*
- *no single evaluator is the best (the harder problems are not found by those who are the most successful)*

Illustration showing which evaluators found which usability problems in a heuristic evaluation of a banking system.
Source: https://www.nngroup.com/articles/how-to-conduct-a-heuristic-evaluation/

How to Conduct Heuristic Evaluation

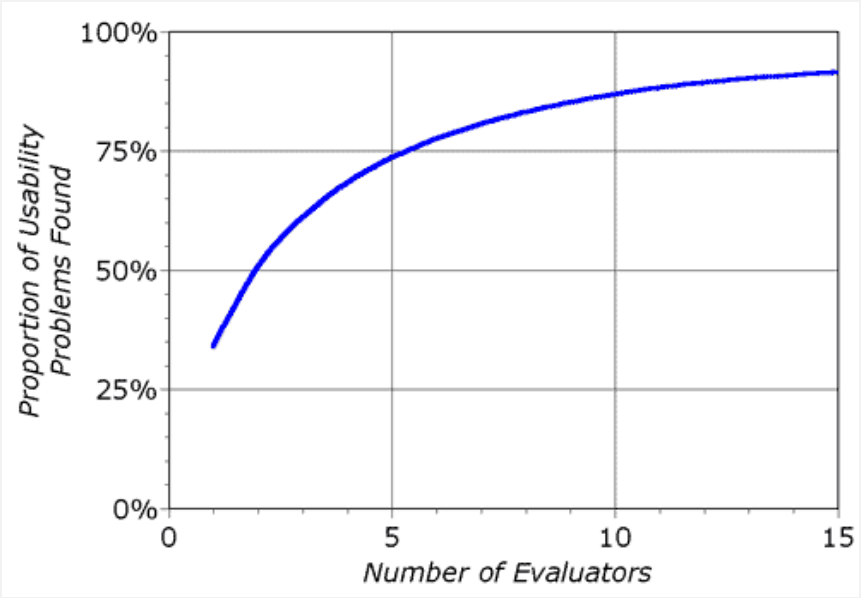
- Have multiple "evaluators" individually go through the interface several times
- Each evaluator inspects various components and compares them with usability principles (heuristics)
  - General rules that seem to describe common properties of usable interfaces (e.g., Nielsen's 10)
  - Any other relevant ones for a specific element/product (e.g., redundancy for critical actions)
- Evaluators communicate and aggregate their findings

- Typically takes 1-2 hours. Split into smaller sessions for large/complicated interfaces

Nielsen, J. 1990. Paper versus computer implementations as mockup scenarios for heuristic evaluation.
Proc. IFIP INTERACT90 Third Intl. Conf. Human-Computer Interaction (Cambridge, U.K., August 27-31), 315-320.

How Many Evaluators?

- Nielsen suggests 3 to 5



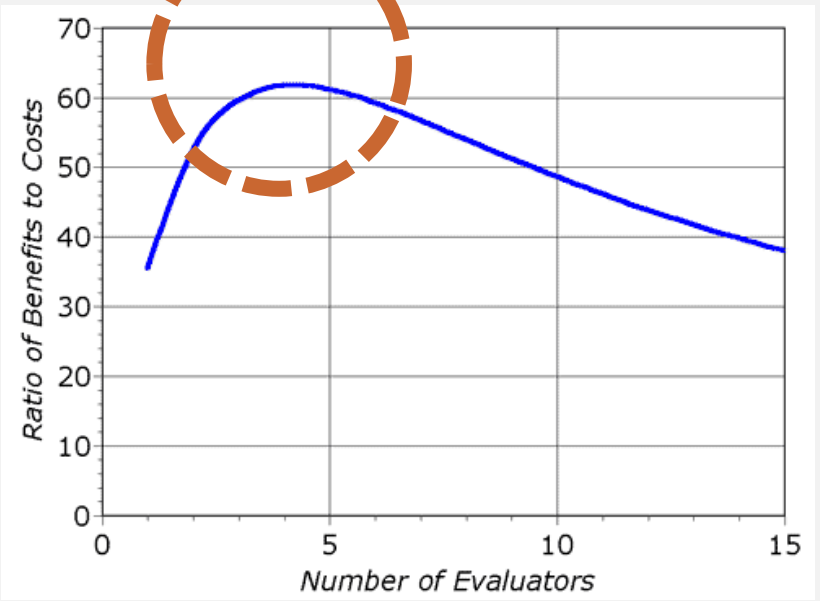
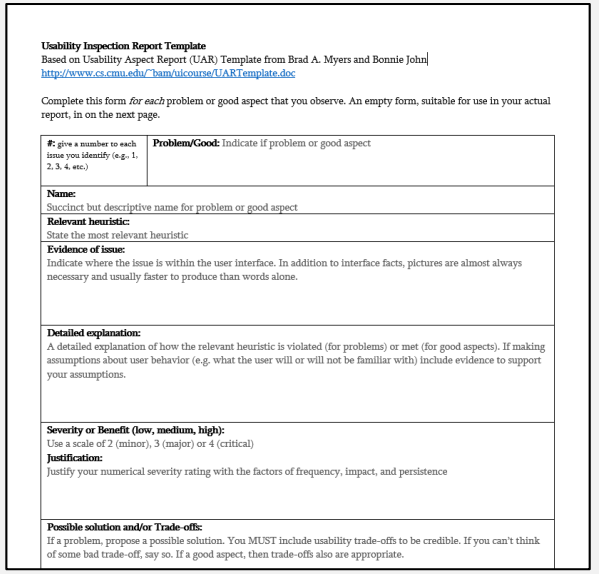


Illustration showing proportion of usability problems found (left) and benefit-to-cost ratio (right) with various number of evaluators. Assuming $15,000 per problem and $4000 + 600n to hire the evaluators.
Source: https://www.nngroup.com/articles/how-to-conduct-a-heuristic-evaluation/

Outputs of Heuristic Evaluation

- A list of usability problems referencing usability principles being violated
- Design advice
  - As an extension of the HE method
  - Typically involves evaluators, observers, and design team representatives
  - Brainstorm possible redesigns to address identified problems
  - Discuss positive aspects of the design (HE doesn't normally has this)



Usability Inspection Report Template
Based on Usability Aspect Report (UAR) Template from Brad A. Myers and Bonnie John
http://www.cs.cmu.edu/~bam/uicourse/UARTemplate.doc

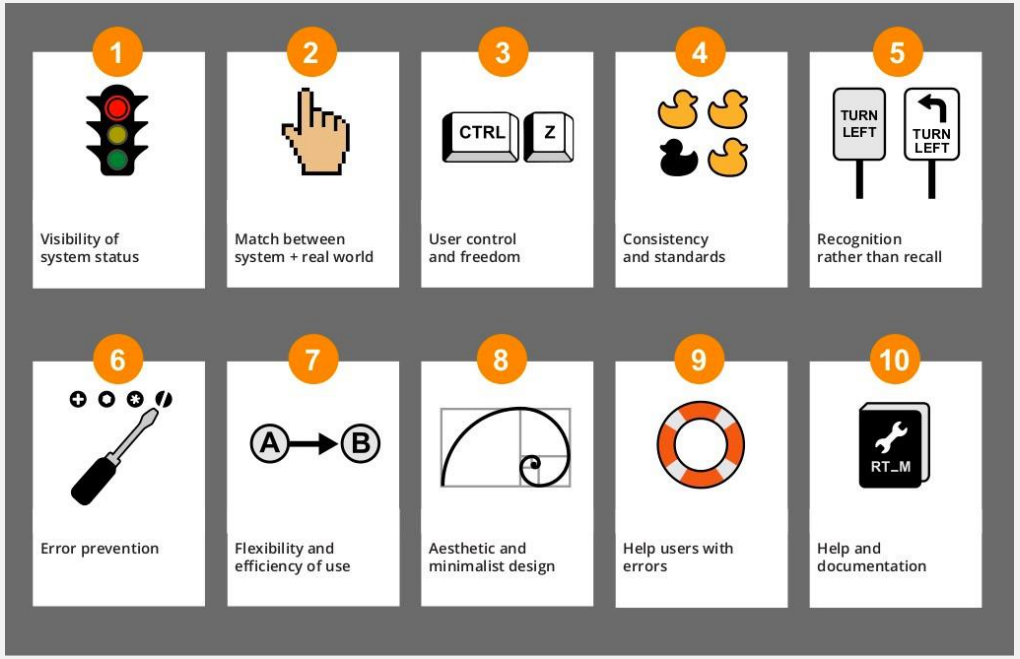
Complete this form *for each* problem or good aspect that you observe. An empty form, suitable for use in your actual report, in on the next page.

#: give a number to each issue you identify (e.g., 1, 2, 3, 4, etc.)	**Problem/Good:** Indicate if problem or good aspect
Name: Succinct but descriptive name for problem or good aspect	
Relevant heuristic: State the most relevant heuristic	
Evidence of issue: Indicate where the issue is within the user interface. In addition to interface facts, pictures are almost always necessary and usually faster to produce than words alone.	
Detailed explanation: A detailed explanation of how the relevant heuristic is violated (for problems) or met (for good aspects). If making assumptions about user behavior (e.g. what the user will or will not be familiar with) include evidence to support your assumptions.	
Severity or Benefit (low, medium, high): Use a scale of 2 (minor), 3 (major) or 4 (critical) **Justification:** Justify your numerical severity rating with the factors of frequency, impact, and persistence	
Possible solution and/or Trade-offs: If a problem, propose a possible solution. You MUST include usability trade-offs to be credible. If you can't think of some bad trade-off, say so. If a good aspect, then trade-offs also are appropriate.	

- Note: it is possible to evaluate the system even when it only exists on paper only (Nielsen 1990)

Nielsen, J. 1990. Paper versus computer implementations as mockup scenarios for heuristic evaluation. Proc. IFIP INTERACT90 Third Intl. Conf. Human-Computer Interaction (Cambridge, U.K., August 27-31), 315-320.

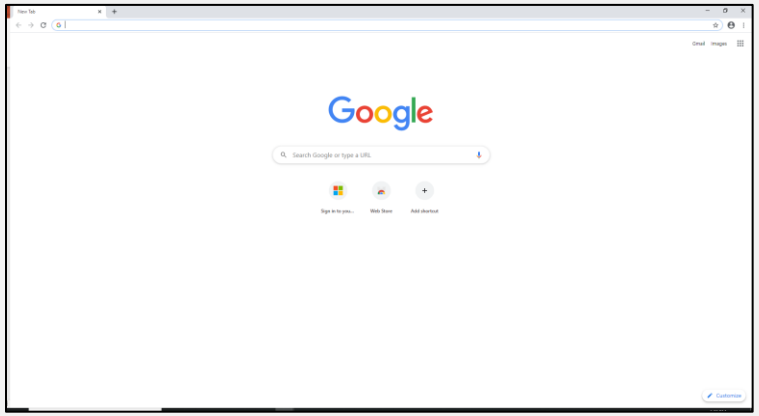
The Nielsen's 10

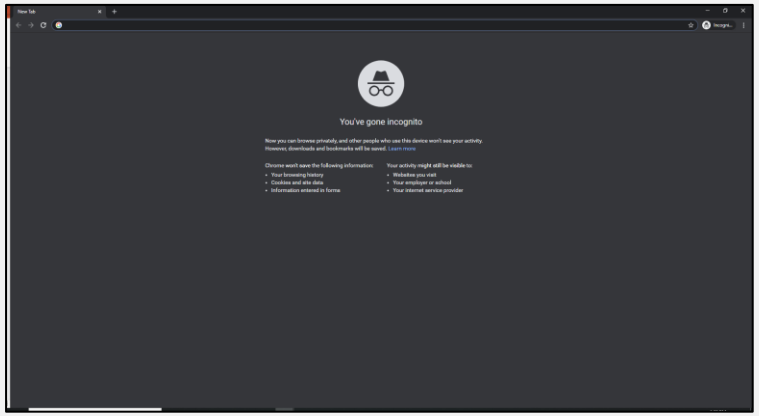


- Visibility of system status
- Match between system + real world
- User control and freedom
- Consistency and standards
- Recognition rather than recall
- Error prevention
- Flexibility and efficiency of use
- Aesthetic and minimalist design
- Help users with errors
- Help and documentation

#1 Visibility of System Status

- Keep users informed about what is going on in the system, through appropriate feedback within reasonable time
  - Example 1: What page they are on and what part of a process (# of steps left)
  - Example 2: In offline mode when connection is lost, or by choice
- Increases transparency, predictability, sense of control, and trust





Indication of normal/incognito modes in Google Chrome

#2 Match between System & The Real World

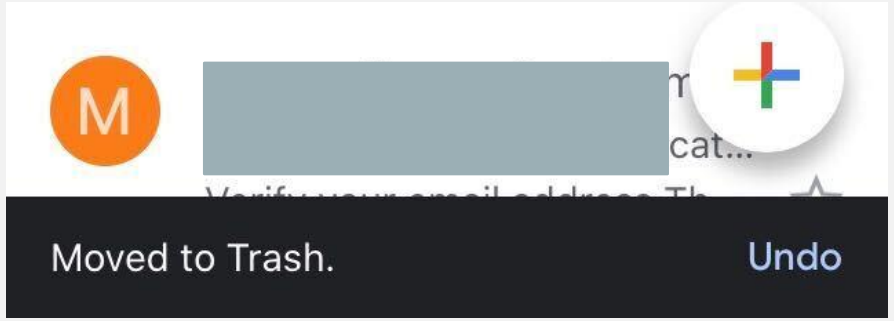
- Speak the users' language that are familiar to the users, follow real-world conventions for showing information, use real-life metaphors to convey meaning
  - Example 1: Say things from users' perspective "You have bought…" instead of "We have sold you"
  - Example 2: Interface components mimic corresponding tools in real-life
- Prevents errors, facilitate understanding, promote feelings of familiarity and care, leading to loyal users

Use icon that reflects real-life counter-part

Use words that are common for shopping

#3 User Control & Freedom

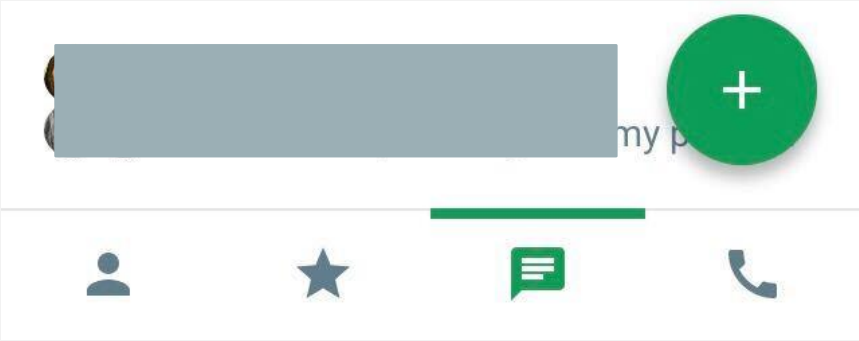
- Expect mistakes, provide an "emergency exit", support undo and redo when possible
  - Example 1: Provide a "start over" option for form filling
  - Example 2: Allow users to remove items from a shopping cart
- Allows flexibility and promote confidence. Even frequent users make mistakes!
  - Note: this heuristic doesn't always refer to mistakes, it can also be ease and availability of control

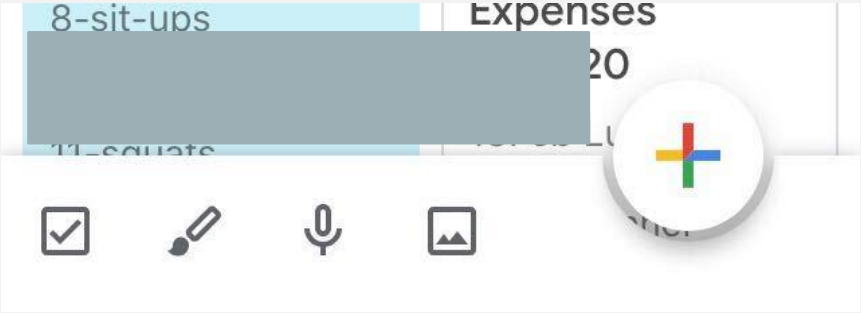


Allowing users to undo email actions in the Gmail app.

#4 Consistency & Standards

- Use the same wordings, actions mean the same thing as frequently as possible
  - For example:
    - Same command results in the same effect (Ctrl+C = copy)
    - Layout in the same website remains the same
- Promotes familiarity (branding), people like consistency (very powerful attribute to usability)
  - Fun read: https://www.nngroup.com/articles/power-law-learning/

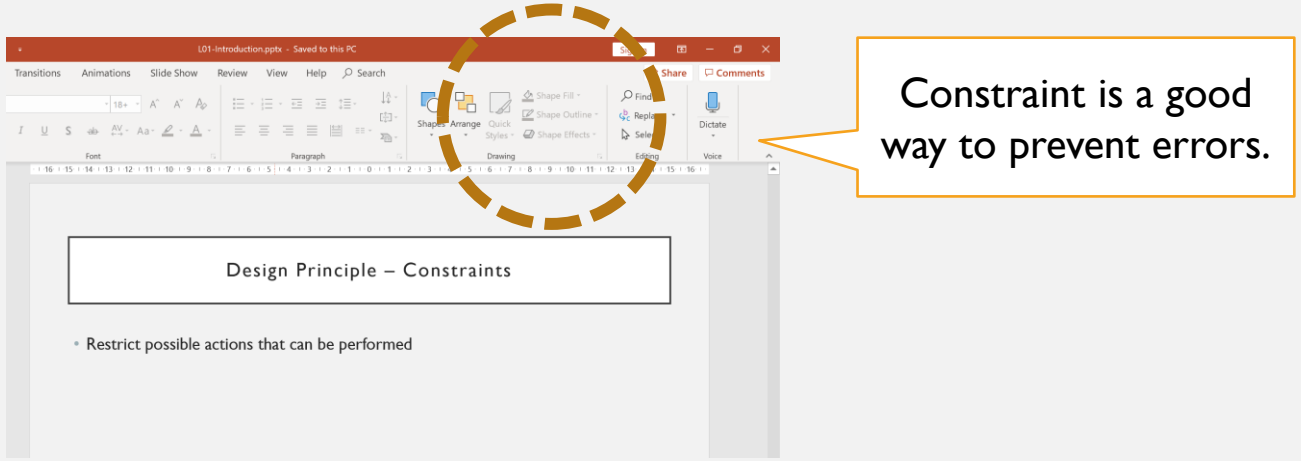




Button for adding content is consistently available at the lower right corner across Google apps.

#5 Error Prevention

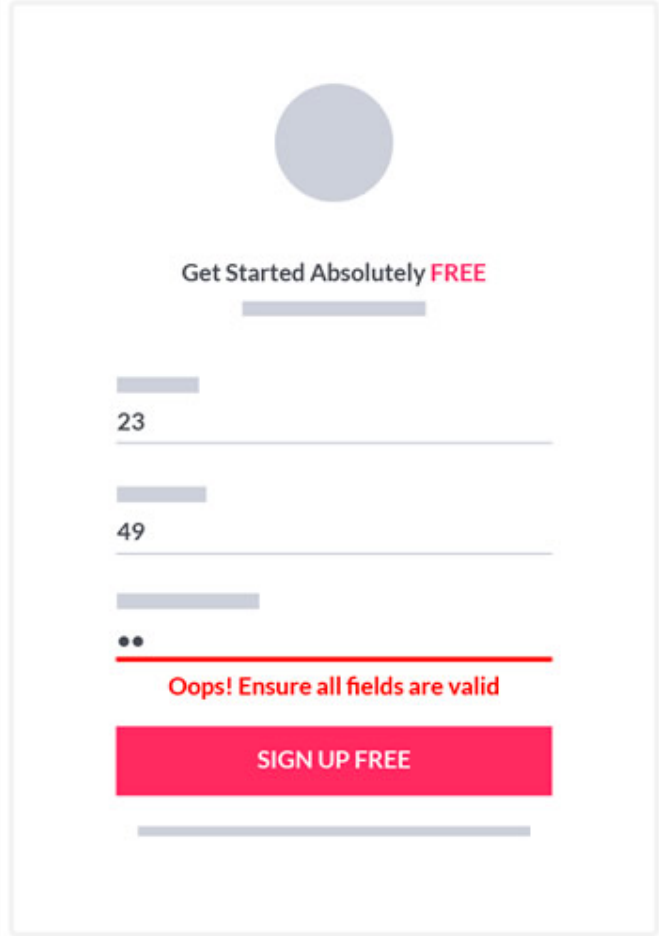
- Eliminate error-prone conditions or check with users using confirmations before committing
  - Example 1: Place the "delete" button further away from the other buttons (prevent "slips")
  - Example 2: Ask the user to confirm their action and suggest double-checking (prevent "mistakes")
- Reduces chances of errors, saves users from unwanted consequences and their time



Constraint is a good way to prevent errors.

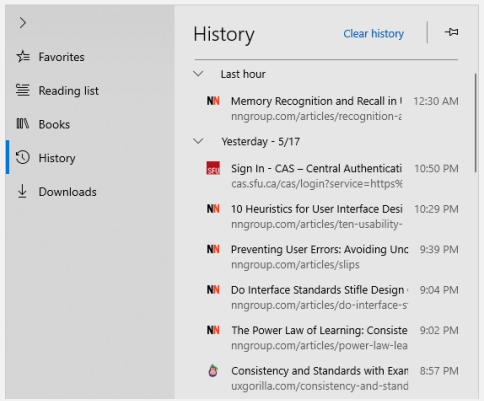
Activity

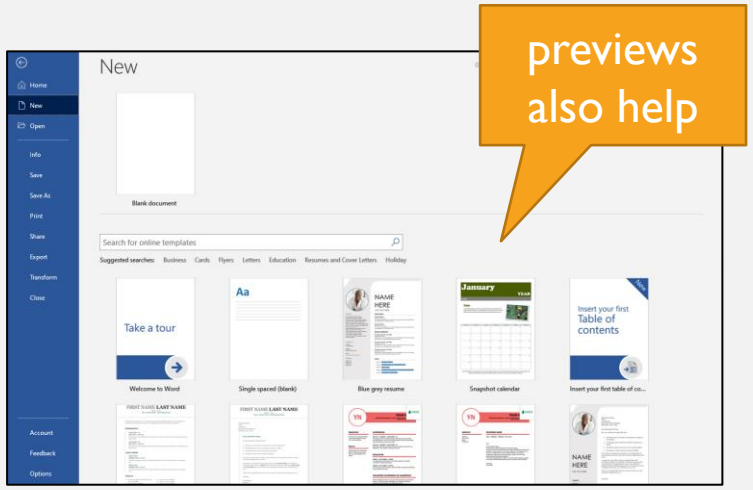
- Chat activity
  – what is wrong with this interface?



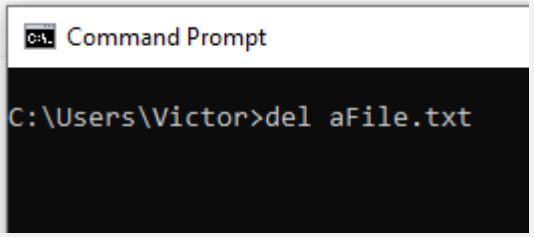
#6 Recognition Rather Than Recall

- Minimize users' memory load by making objects, actions, and option visible
  - Example 1: Provide history of actions or sites visited
  - Example 2: Auto-fill of previously inputted emails or addresses
- Reliefs users from having to remember information from one part of the dialogue to another



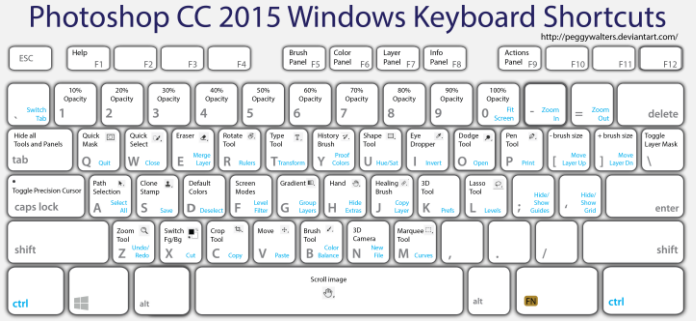


VS



#7 Flexibility & Efficiency of Use

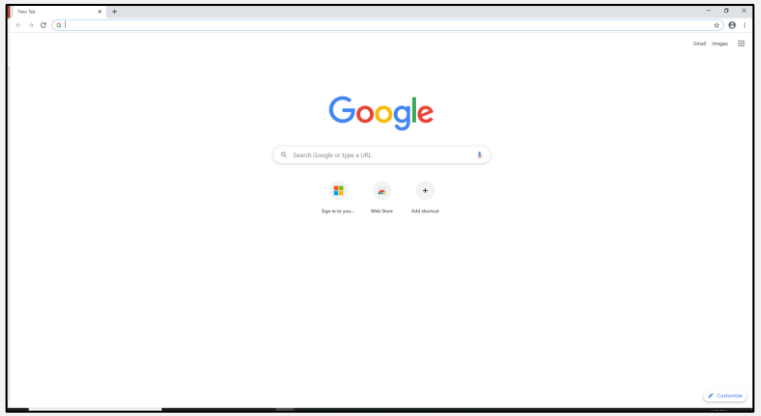
- Provide multiple ways of different efficiencies to do the same thing, including accelerators that speeds up process, or provide defaults
  - Example 1: Multiple ways to copy highlighted text – Edit > Copy, right-click > Copy, and Ctrl+C
  - Example 2: Macros/scripts in Photoshop to apply the same effects to multiple images
- Allows users of various experiences with the interface to do things in their own ways



Source: left-https://theblog.adobe.com/photoshop-cc-2015-and-fuse-cc-preview-available-today/
right-https://www.deviantart.com/peggywalters/art/Photoshop-Shortcuts-593860178

#8 Aesthetic & Minimalist Design

- Keep content focus on the essential through good graphics design and colour choice
  - Example 1: Use alignments and other layouts to compartmentalize content
  - Example 2: Infrequently used functions are hidden
- Increases signal-to-noise ratio, draws users' attention to the right place at the right time
  - Signal-to-noise ratio: https://www.nngroup.com/articles/signal-noise-ratio/



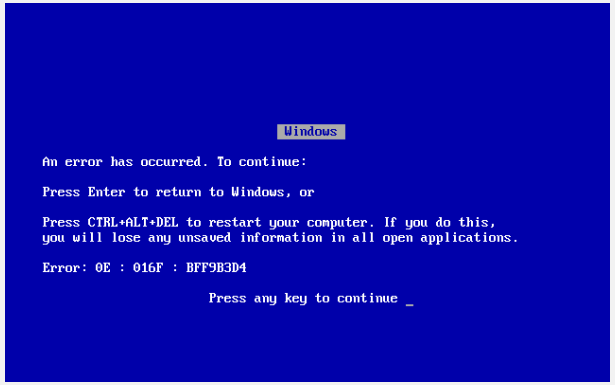
The main focus is the search bar in the Google search engine.

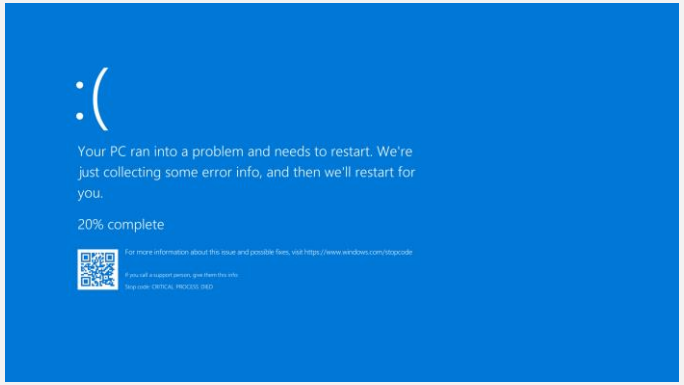
Notes on Aesthetics & Minimalistic Design

- Do not rely on colours for improving readability or to design affordance
  - Some people are colour blind (~8% of males, 0.6% of females)
  - Some people have dyslexia (~15% of people)
- Some times minimal isn't the best approach
  - Doing too much will make things confusing and ambiguous
  - Redundancy or multiple access could be a useful feature
  - Content-rich vs Cluttered: https://www.webdesignerdepot.com/2009/12/when-minimalism-backfires-when-too-little-is-not-enough/

#9 Help Users with Errors

- Indicate errors in ways to help users recognize, diagnose, and recover from errors
  - Example 1: Most websites now provide a custom "page not found" error and offer to bring the user back
  - Example 2: Systems will warn users for empty fields in the online form
- Reduces stress and promotes confidence and loyalty





Modern Windows' BSOD is providing more information about the error.

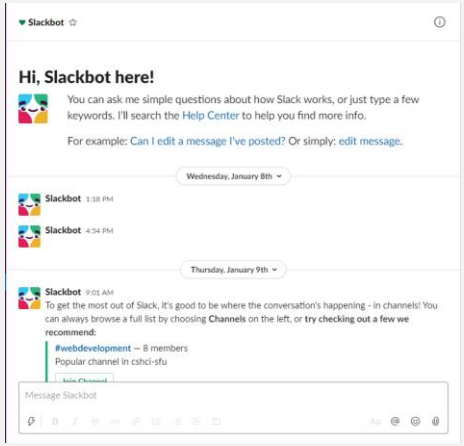
Source: left-By The original uploader was Akhristov at English Wikipedia. - Transferred from en.wikipedia to Commons by TFCforever., Public Domain, https://commons.wikimedia.org/w/index.php?curid=8782066
right-By MitrandirLK - Own work, Public Domain, https://commons.wikimedia.org/w/index.php?curid=48162543

Notes on Help Users with Errors

- Blame the system, not the user
  - "Unrecognized" vs "illegal" command
- Use plain language
  - No "error 0x00000000234", users are probably under a lot of stress!
- Levels of messages
  - Error < explanation < solution

#10 Help & Documentation

- Provide user assistance at appropriate time in the interaction, with easily searchable content in concrete steps
  - Example 1: Help pages in software and websites
  - Example 2: "Getting started", chatbots, tooltips, related help throughout different phases of use
- Allows users to learn on their own and become an expert by themselves



Modern software (e.g., Slack) has a built-in conversational help tool.

Extended Readings/Watchings of the Nielsen's 10

- Visibility of System Status: https://www.nngroup.com/articles/visibility-system-status/
- Match between System + The Real World: https://www.nngroup.com/articles/match-system-real-world/
- User Control & Freedom: https://uxgorilla.com/user-control-and-freedom/
- Consistency & Standards: https://uxgorilla.com/consistency-and-standards/
- Error prevention: https://www.nngroup.com/articles/slips/
- Recognition > Recall: https://www.nngroup.com/articles/recognition-and-recall/
- Aesthetic & Minimalist Design: https://uxgorilla.com/aesthetic-and-minimalist-design/
- Help Users with Errors: https://uxgorilla.com/help-users-recognise-diagnose-and-recover-from-errors/
- Help & Documentation: https://uxgorilla.com/help-and-documentation/

- Lectures with Dr. Scott Klemmer: https://www.youtube.com/playlist?list=PLVtu1bDQijari7LfHOoSTdcpbWIkwZWIA

Summary

- Evaluating Interfaces
  - Why? What? Where? When?
- Heuristic Evaluation
  - Expected output
  - Nielsen's Heuristics

Post-Lecture Activity

- Read/watch these (and those in the slides)
  - Chapters 14 & 15 of ID-Book: Introducing Evaluation & Evaluation Studies
  - Thinking Aloud: The #1 Usability Tool
    https://www.nngroup.com/articles/thinking-aloud-the-1-usability-tool/
  - Usability evaluation and analysis
    https://www.hotjar.com/usability-testing/evaluation-analysis/

- Think about these as exercises:
  - Suppose you are designing the interface for a new smartwatch for fitness. What will you test? Where will you test it? When in the development process will you test it?
  - Which heuristics do you think are the more important ones for a new user of the interface?

In-Class Activity

- As a group
  - Go to https://snap.sfu.ca/
  - Explore what it does
  - Evaluate its interface using Nielsen's Heuristics
    - Find at least 2 features that demonstrate 2 different heuristics
      - E.g., the use of recognizable icons at the bottom showing different links helps users to know what they do (Recognition rather than recall)
    - Find at least 2 features that violate 2 different heuristics
      - E.g., The app logo does not do anything, while typically the icon at the top-left brings the user back to "home" (Consistency & standard)