

User Interface Design & Evaluation

Evaluation

Overview



- Scenario-based design
- Prototyping
- Evaluation

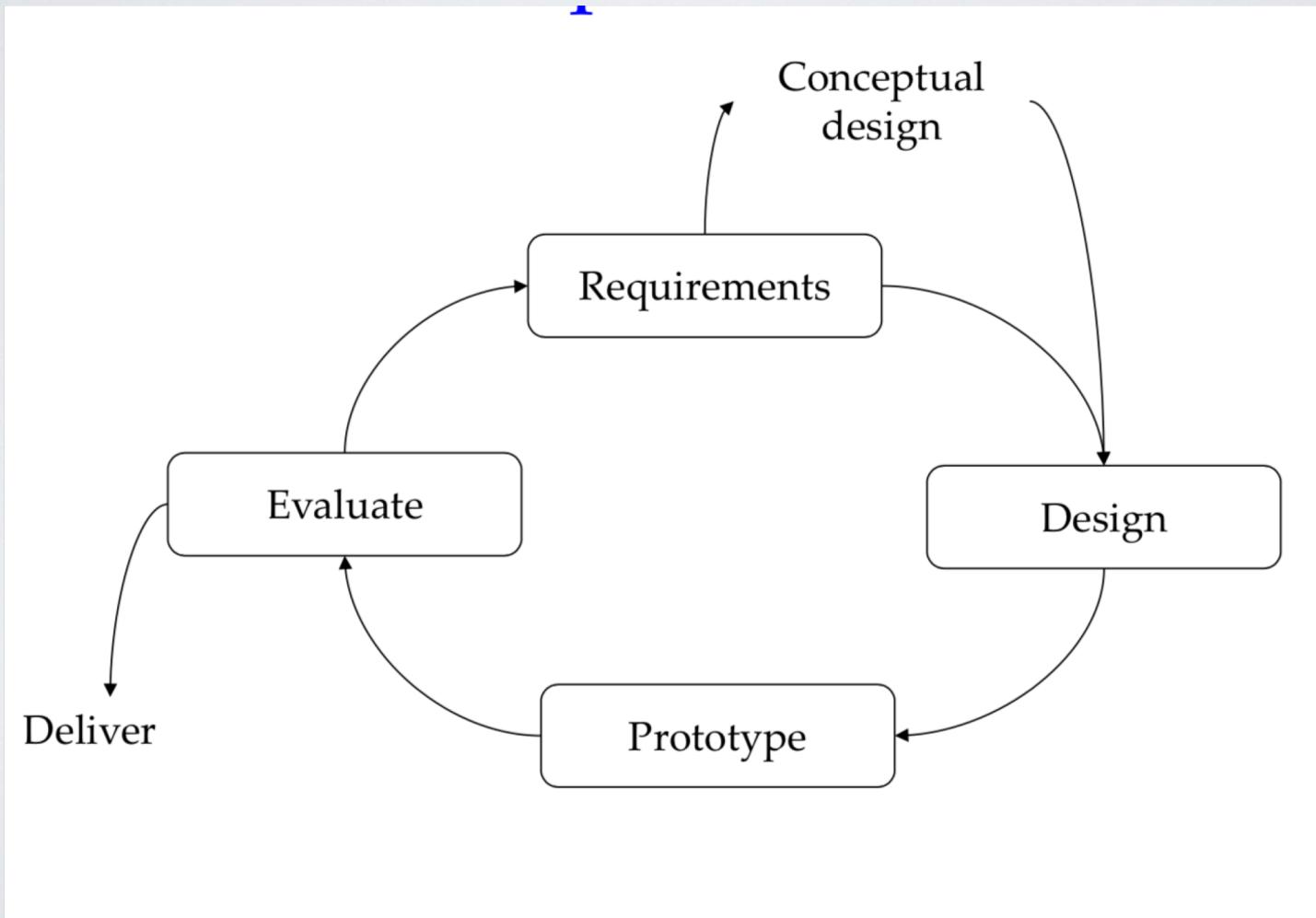
Overview

- Why, what and when to evaluate
- How to evaluate
 - Heuristic evaluation
 - Walkthroughs
 - Think Aloud
 - SUS

WHAT IS EVALUATION?

- Does/will the system work?
- Is it usable by your users?
- What else needs to be done?
- Is it a *good* system?

ITERATIVE DEVELOPMENT



REQUIREMENTS EVALUATION

Evaluation leads to needs
Evaluate with your system
Overlapping methods

WHY EVALUATE?

- Does the item achieve what it is supposed to?
- What works in theory might not work in practice
- User experience is subjective

WHAT TO EVALUATE

- What artefacts to evaluate
 - Anything
- What aspects to evaluate
 - Whatever is relevant

WHEN TO EVALUATE

- Depends on the development process
- Depends on what is being evaluated
- Detecting problems early results in least wasted time and money

TYPES OF EVALUATION

- Field studies
 - Observe, interview, questionnaire
- Analytic evaluation
 - Heuristic evaluation, walkthroughs, models
- Usability testing
 - Lab-based evaluation, eyetracking

HOW TO EVALUATE

- Evaluation with users in a natural setting
- Evaluation with users in an artificial setting
- Evaluation without users (analytic evaluation)

EVALUATION IN A NATURAL SETTING

- Evaluate how users interact with artefact “in the wild”
- Only really possible with a very high fidelity prototype
- Exactly the same techniques as used for gathering requirements

EVALUATION IN AN ARTIFICIAL SETTING

- Evaluate how users perform tasks in a lab setting
- Monitoring equipment available
 - Cameras, logging, eye-tracking etc.
- Loss of context around activities
- See the second half of HACS for details

ANALYTIC EVALUATION

- Otherwise known as inspection or review
- Analysis of the system by an experts, other members of the team or users
- Two major techniques
 - Heuristic evaluation
 - Walkthroughs

HEURISTIC EVALUATION

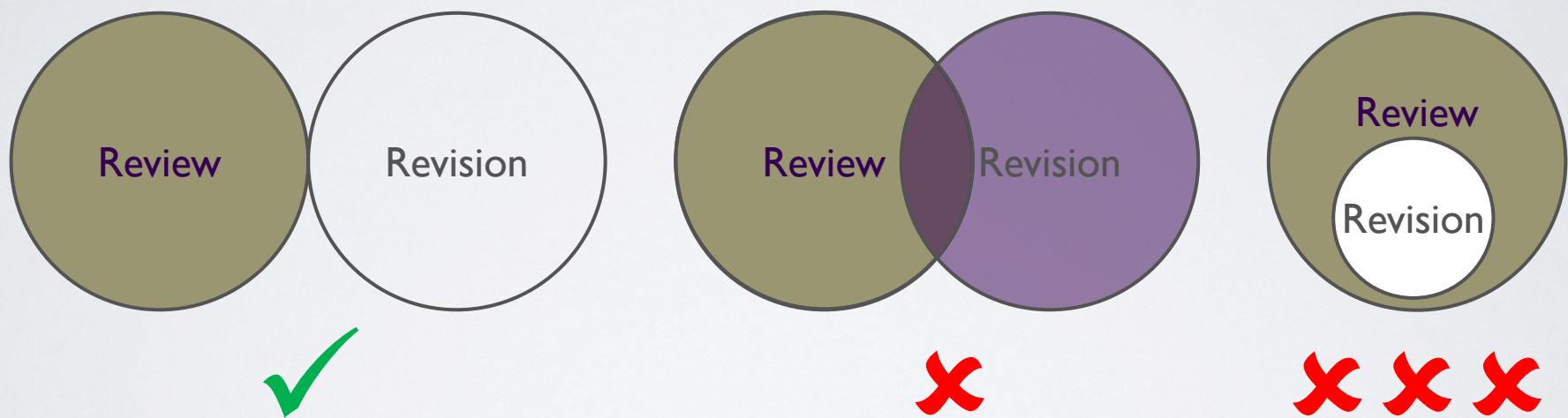
- Conducted by experts
- Artefact is examined to see how well it meets set of criteria
 - e.g. Nielson's usability heuristics
 - Can evaluate anything in this way, as long as there are suitable criteria

- Visibility of system status
- Match system and real world
- User control and freedom
- Consistency and standards
- Error prevention
- Recognition rather than recall
- Flexibility and efficiency of use
- Aesthetic and minimalist design
- Help users (recover from errors)

HEURISTIC EVALUATION PROCESS

- More experts result in more problems being found
- An evaluation has the structure:
 1. Briefing session
 2. Evaluation period
 3. Debriefing session

REVIEW AND REVISION



HEURISTIC EVALUATION

- Work through using heuristics
- Vague
- Need experts
- Need several but NOT lots

HEURISTIC EVALUATION

- Take a phone, tablet or laptop
- Evaluate any simple program (e.g. calendar, messaging, alarm clock) using these heuristics
 - Match between system and real world
 - Error prevention
 - Flexibility and efficiency of use

HEURISTIC EVALUATION

- Is it enough?
- Not for complex tasks
- Mismatch of severity
- Depends on experts
- Users are still different!

HEURISTIC EVALUATION

EXERCISE:

Evaluate someone's mobile phone

Note issues, problems, things to consider

Visibility of system status

Match between the system and the real world Consistency and standards

Flexibility and efficiency of use

Aesthetic and minimalist design

WALKTHROUGHS

- Conducted by a team
- Walkthrough the actions that must be performed to accomplish a specific task
- Consider the behaviour and understanding of a target user at each step

QUESTIONS

- I. Will the user know what to do at this step?
2. If the user does the right thing, will they know they did the right thing, and are making progress towards their goal?

TAKE HOME LESSONS

- Design is about function, not just looks
- Always think about the user, their goals and their abilities
- Context matters

Heuristics

OVERVIEW

- Analytic evaluation
- Nielson's usability heuristics

TYPES OF EVALUATION

- Evaluation with users in a natural setting
- Evaluation with users in an artificial setting
- Evaluation without users (analytic evaluation)

ANALYTIC EVALUATION

- Otherwise known as inspection or review
- Analysis of the system by an experts, other members of the team or users
- Two major techniques
 - Heuristic evaluation
 - Walkthroughs

HEURISTICS

- Rules that are generally true but not always
- Trade accuracy and precision for simplicity
 - I before E, except after C
- Generally useful, but not true for ‘either’, ‘leisure’, ‘science’ etc.

DESIGN HEURISTICS

- Guidelines on good design
- Lots exist, including
 - Nielsen's 10 usability heuristics
 - Shneiderman's 8 golden rules
 - Tog's first principles of interaction design
 - Petrie and Power's web usability heuristics

NIELSEN'S USABILITY HEURISTICS

1. Visibility of system status
2. Match between system and the real world
3. User control and freedom
4. Consistency and standards
5. Error prevention

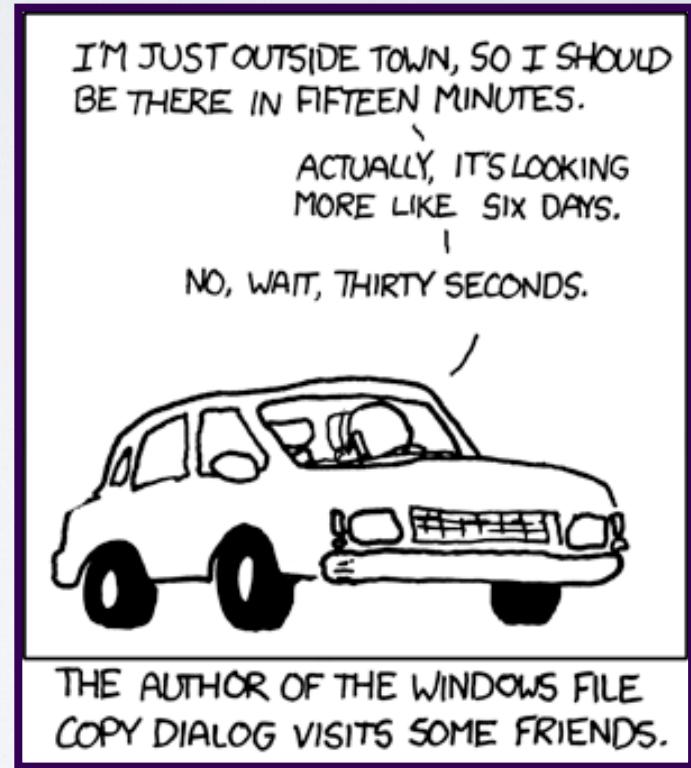
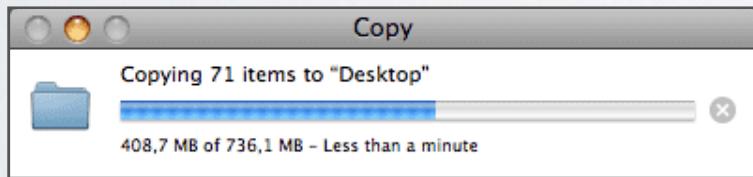
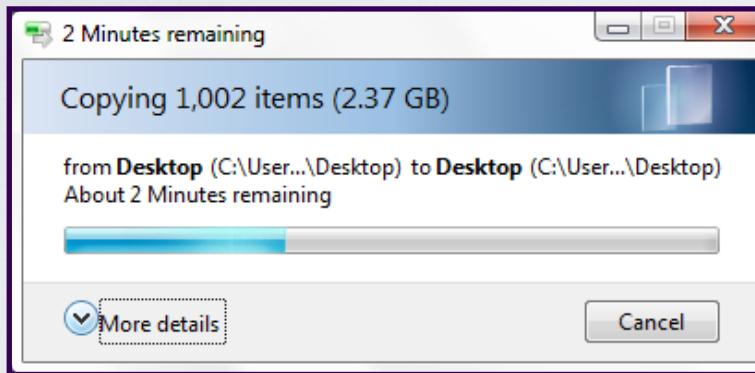
NIELSEN'S USABILITY HEURISTICS

6. Recognition rather than recall
7. Flexibility and efficiency of use
8. Aesthetic and minimalist design
9. Help users recognise, diagnose and recover from errors
10. Help and documentation

I. VISIBILITY OF SYSTEM STATUS

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

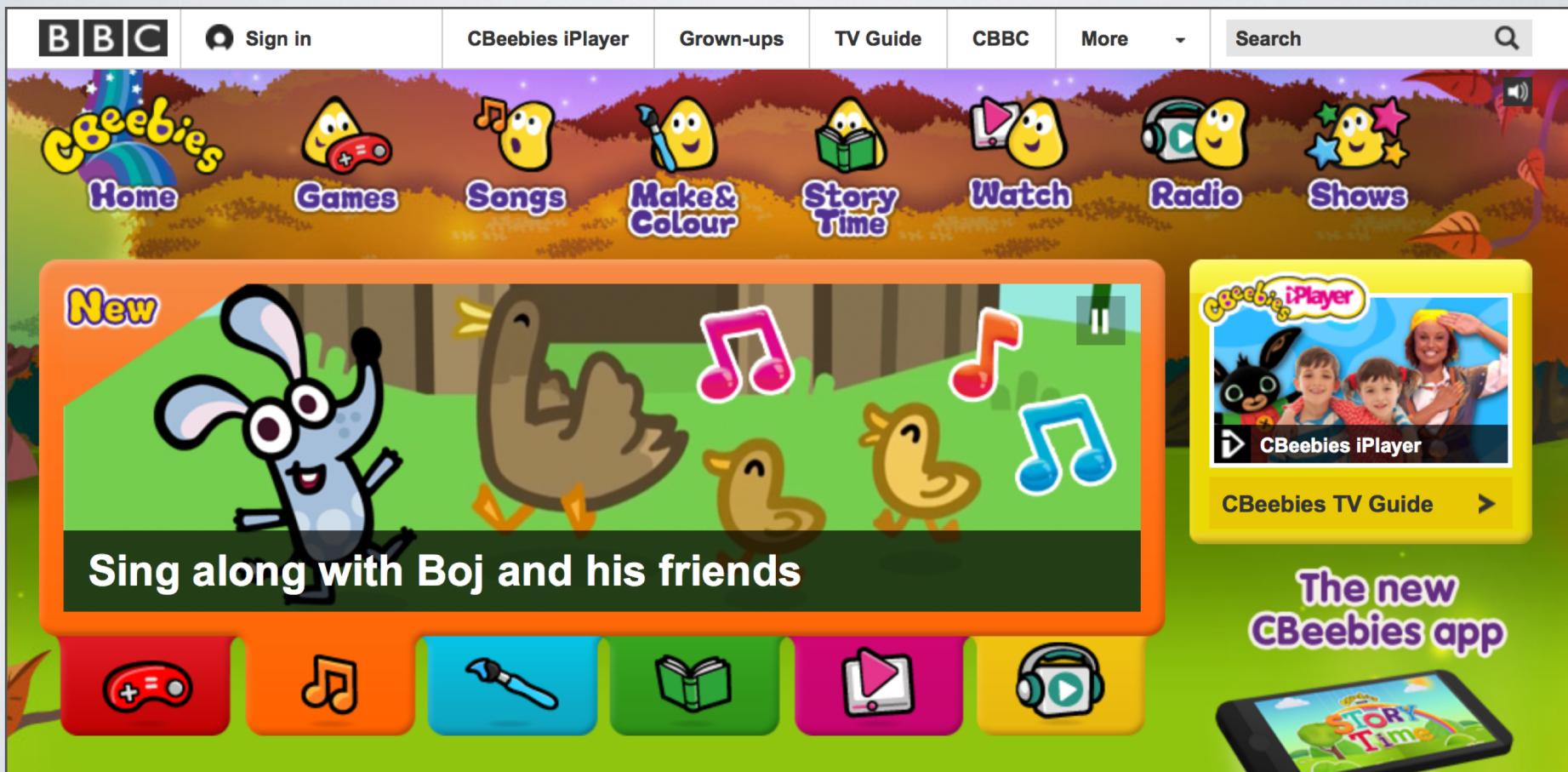
I. VISIBILITY OF SYSTEM STATUS



2. MATCH BETWEEN SYSTEM AND THE REAL WORLD

- 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.

2. MATCH BETWEEN SYSTEM AND THE REAL WORLD



2. MATCH BETWEEN SYSTEM AND THE REAL WORLD



400. That's an error.

Your client has issued a malformed or illegal request. That's all we know.



3. USER CONTROL AND FREEDOM

- 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.

3. USER CONTROL AND FREEDOM

https://planon.york.ac.uk/SelfService/pfnr.dll/res?DEF=7&SID=1&PID=ListPage&CPIID=SearchPage&PA=InitPage&RM=Main&LBLM=fst&SEAR

Rooms matching your criteria

Page: [Show page](#) [Search page](#) \ Reservation unit list page

Help

Click on your required start time below (each of the light blue blocks equals 15 minutes)

Please note that teaching rooms should be booked from quarter past the hour.

If the message **No results found** appears below, try another search, but with fewer facilities selected.

Avoid using your browser **Back** button to return you to your original selection. Instead use the **Search page** link above.

Click the tick box at the bottom of the screen if the room you are expecting to see is not displayed. This will show any rooms whe

To view information on a room click the 'i' button

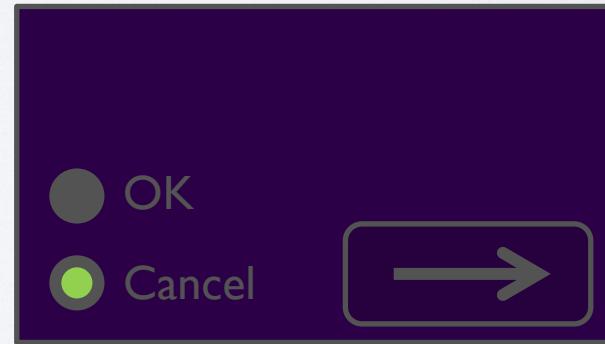
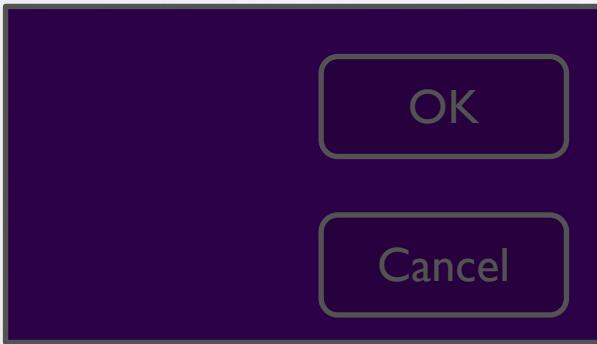
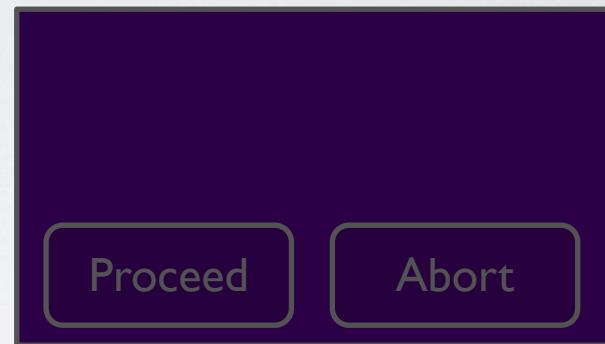
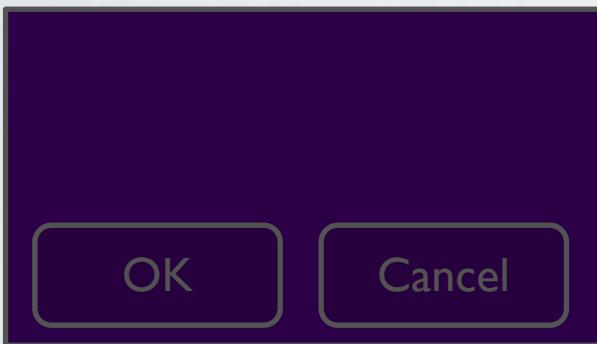
Name:	AEW/003 Lecture Room
Comment:	During vacations this room is only available if your request is made within two weeks of the date of your booking.
Building opening times:	ALCUIN EAST WING - Mon-Fri 08:15 - 22:00
Number of people:	100

7:00 8:00 9:00 10:00 11:00

4. CONSISTENCY AND STANDARDS

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

4. CONSISTENCY AND STANDARDS

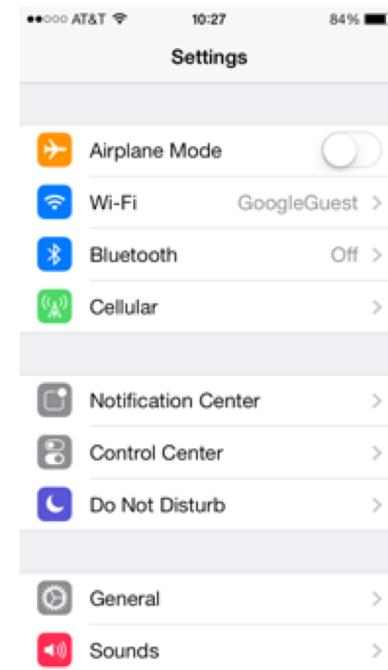
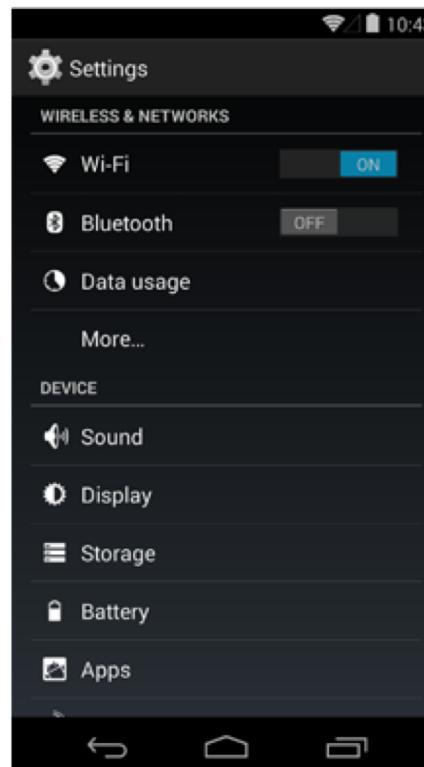


4. CONSISTENCY AND STANDARDS

Don't use right-pointing carets on line items

A common pattern on other platforms is the display of right-pointing carets on line items that allow the user to drill deeper into additional content.

Android does not use such indicators on drill-down line items. Avoid them to stay consistent with the platform and in order to not have the user guess as to what the meaning of those carets may be.

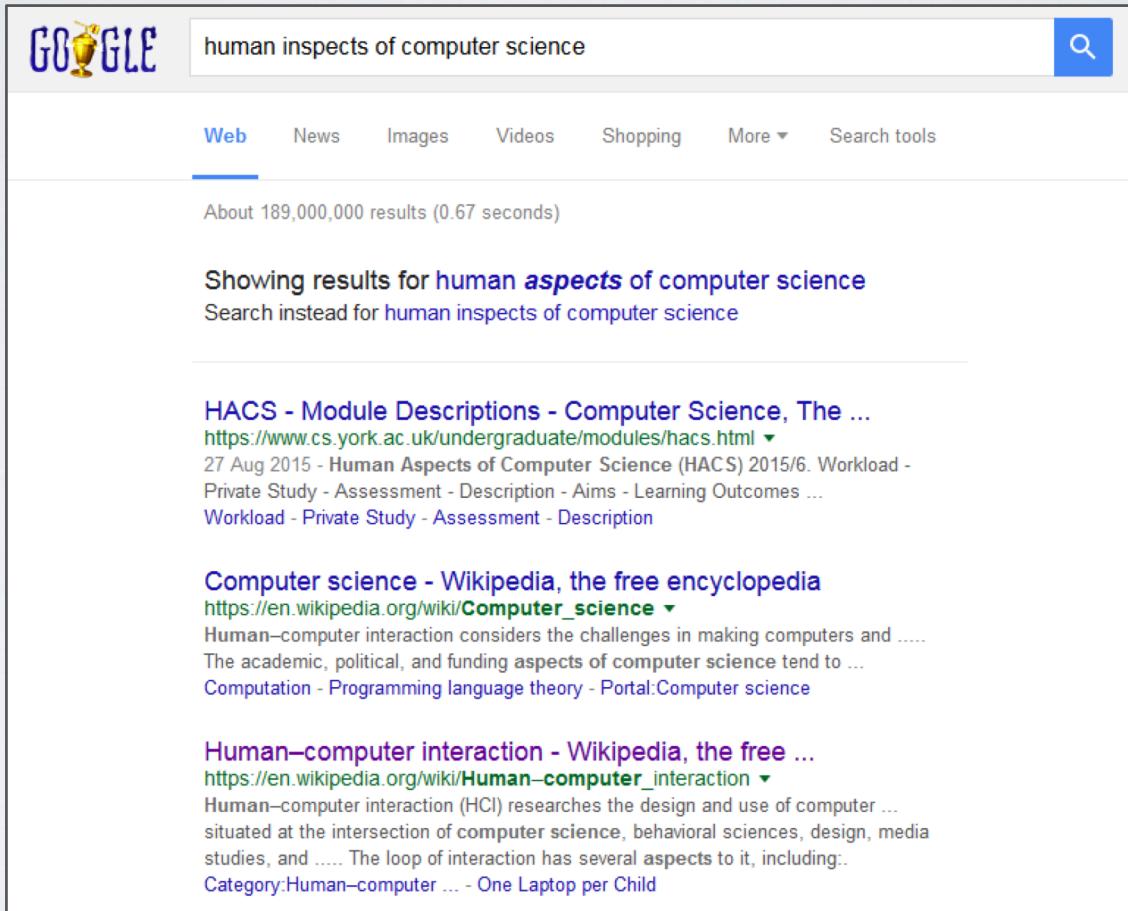


Android settings without right-pointing carets in line items vs. iOS settings.

5. ERROR PREVENTION

- 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.

5. ERROR PREVENTION



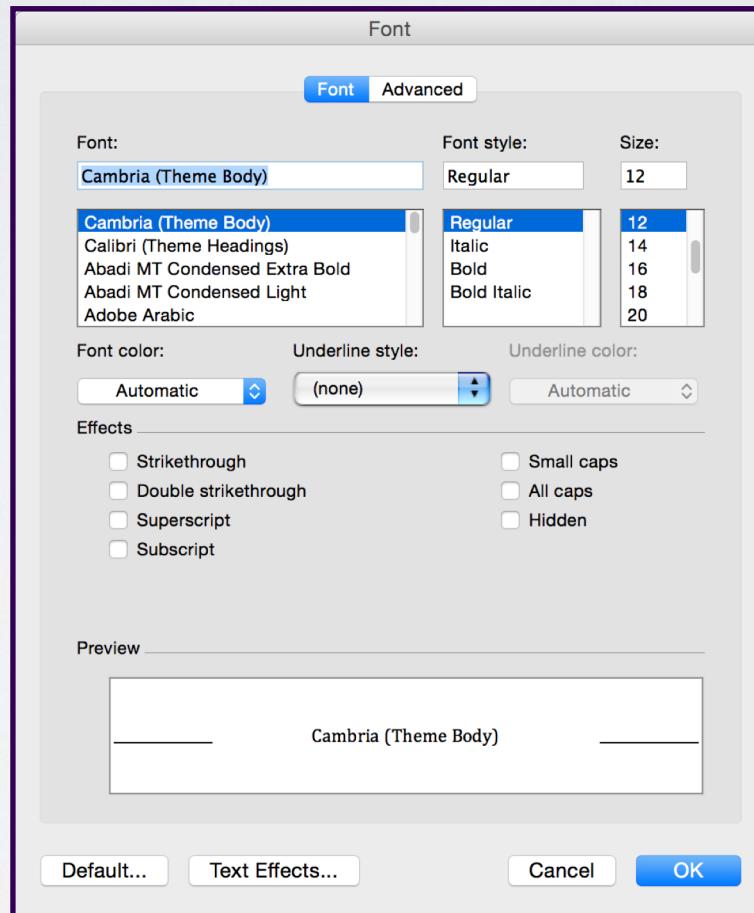
A screenshot of a Google search results page. The search query is "human aspects of computer science". The results are as follows:

- HACS - Module Descriptions - Computer Science, The ...**
<https://www.cs.york.ac.uk/undergraduate/modules/hacs.html> ▾
27 Aug 2015 - Human Aspects of Computer Science (HACS) 2015/6. Workload - Private Study - Assessment - Description - Aims - Learning Outcomes ...
Workload - Private Study - Assessment - Description
- Computer science - Wikipedia, the free encyclopedia**
https://en.wikipedia.org/wiki/Computer_science ▾
Human–computer interaction considers the challenges in making computers and The academic, political, and funding aspects of computer science tend to ...
Computation - Programming language theory - Portal:Computer science
- Human–computer interaction - Wikipedia, the free ...**
https://en.wikipedia.org/wiki/Human-computer_interaction ▾
Human–computer interaction (HCI) researches the design and use of computer ... situated at the intersection of computer science, behavioral sciences, design, media studies, and The loop of interaction has several aspects to it, including:
Category:Human–computer ... - One Laptop per Child

6. RECOGNITION RATHER THAN RECALL

- 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.

6. RECOGNITION RATHER THAN RECALL



6. RECOGNITION RATHER THAN RECALL

A screenshot of a web browser window showing search results for the query "hacs". The results are listed in a vertical scrollable list. The first result is a link to the CS Staff Intranet at the University of York, followed by several links related to the HACS module pages at the University of York. One result is highlighted with a blue background, indicating it is the current page being viewed.

- Human Aspects of Computer Science - CS Staff Intranet, The University of York
<https://www.cs.york.ac.uk/modules/hacs.html>
- HACS Module pages, Department of Computer Science, University of York, UK
www-module.cs.york.ac.uk/hacs/
- HACS Module pages, Department of Computer Science, University of York, UK
www-module.cs.york.ac.uk/hacs/Assessment1/index.html
- HACS Module pages, Department of Computer Science, University of York, UK
[file:///C:/Users/mbartlett/Dropbox/Teaching Fellow/HACS/hacs/index.html](file:///C:/Users/mbartlett/Dropbox/Teaching%20Fellow/HACS/hacs/index.html)
- www-module.cs.york.ac.uk
www-module.cs.york.ac.uk/hacs
- Index of /hacs/Lectures
www-module.cs.york.ac.uk/hacs/Lectures/

6. RECOGNITION RATHER THAN RECALL

Complete this form to reset your LastPass master password

Email

Old Password

New Master Password

Re-Enter New Master Password

New Password Reminder (Optional)

SAVE MASTER PASSWORD

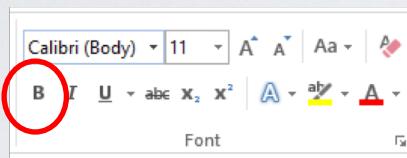
7. FLEXIBILITY AND EFFICIENCY OF USE

- 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.

7. FLEXIBILITY AND EFFICIENCY OF USE

- Allow novice users to interact with the system easily, while allowing expert users to interact with it efficiently. Give users flexibility in how they use the system and allow them to customise it.

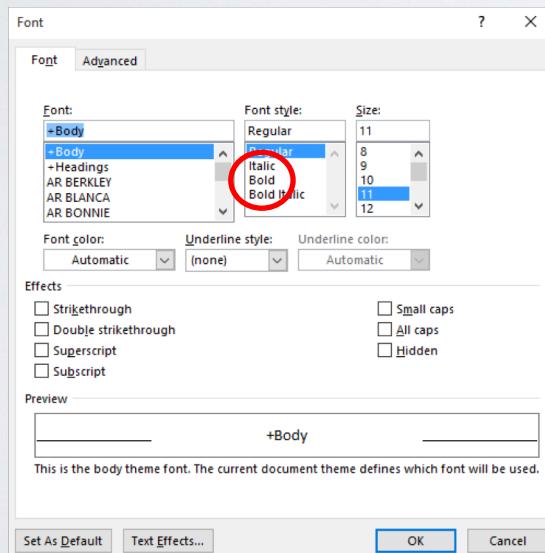
7. FLEXIBILITY AND EFFICIENCY OF USE



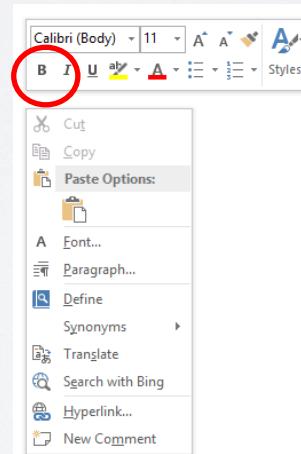
Toolbar button



Apply style



Font dialog

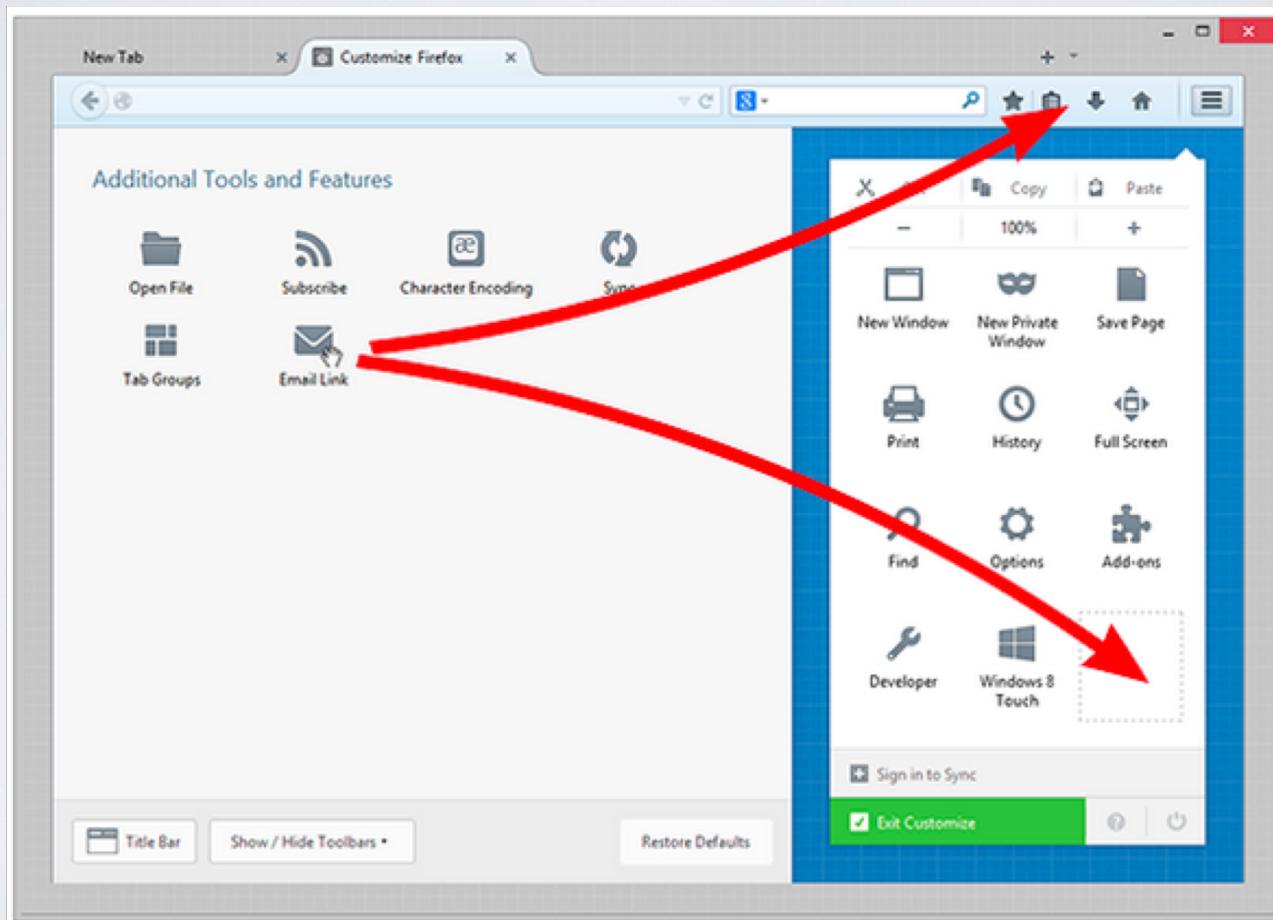


Context menu

Ctrl-B

Keyboard shortcut

7. FLEXIBILITY AND EFFICIENCY OF USE



8. AESTHETIC AND MINIMALIST DESIGN

- 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.

8. AESTHETIC AND MINIMALIST DESIGN

WIKIPEDIA
The Free Encyclopedia

Main page
Contents
Featured content
Current events
Random article
Donate

Interaction
Help
About Wikipedia
Community portal
Recent changes
Contact Wikipedia

Toolbox

Print/export

Languages
Français
Bahasa Melayu
Русский
Tiếng Việt
中文

Article Discussion Read Edit View history Search

Please read:
A personal appeal from
Wikipedia founder Jimmy Wales

Read Now



Usage share of web browsers

From Wikipedia, the free encyclopedia

The **usage share of web browsers** is the percentage of visitors to a group of websites that use a particular web browser. For example, when it is said that Firefox has 30% usage share, it means that some version of Firefox is used by 30% of visitors that visit a given set of sites.^[1]

In Germany, Firefox is the most popular browser with 60% usage.^[2] In China Internet Explorer has a 89% usage share.^[3]

Contents [hide]

1 Accuracy

1.1 Overestimation

1.2 Underestimation

1.3 Regional variations

2 Summary table

3 Historical usage share

3.1 StatCounter (July 2008 to present)

Source	Internet Explorer	Firefox	Chrome	Safari	Opera	Mobile browser
Clicky	47.74%	29.67%	12.69%	8.46%	1.26%	
Stat Counter	49.21%	31.24%	12.39%	4.56%	2.00%	3.81%
W3Counter	41.6%	29.9%	11.9%	5.5%	1.9%	
Wikimedia	44.72%	29.67%	9.71%	5.57%	3.48%	4.70%
Median	46.23%	29.79%	12.15%	5.54%	1.95%	4.26%

Browser usage on Wikimedia - Oct 2010

9. HELP USERS RECOGNISE, DIAGNOSE AND RECOVER FROM ERRORS

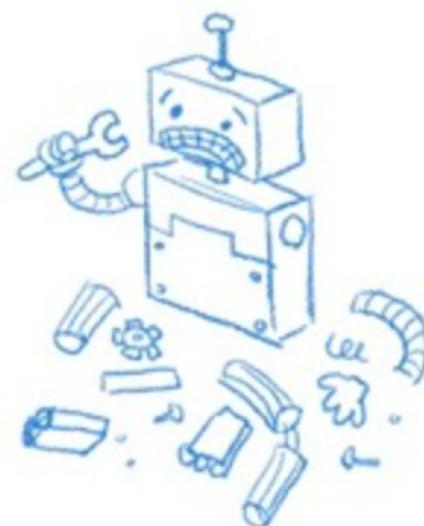
- Error messages should be expressed in plain language (no codes), precisely indicate the problem, and constructively suggest a solution.

9. HELP USERS RECOGNISE, DIAGNOSE AND RECOVER FROM ERRORS

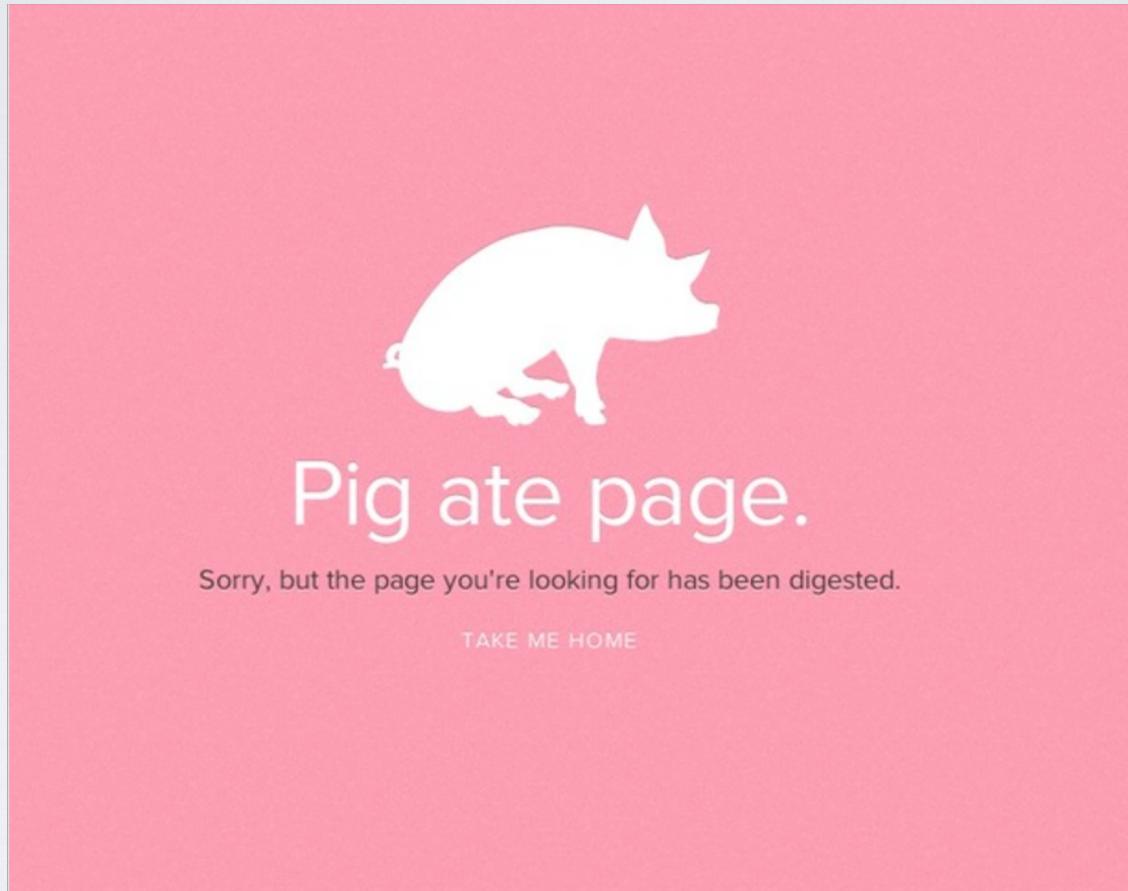


400. That's an error.

Your client has issued a malformed or illegal request. That's all we know.



9. HELP USERS RECOGNISE, DIAGNOSE AND RECOVER FROM ERRORS



9. HELP USERS RECOGNISE, DIAGNOSE AND RECOVER FROM ERRORS



Error

404 – File not found

<http://wikimediafoundation.org/404ed>

We could not find the above page on our servers.

Did you mean to type

<http://wikimediafoundation.org/wiki/404ed>? You will be automatically redirected there in five seconds.

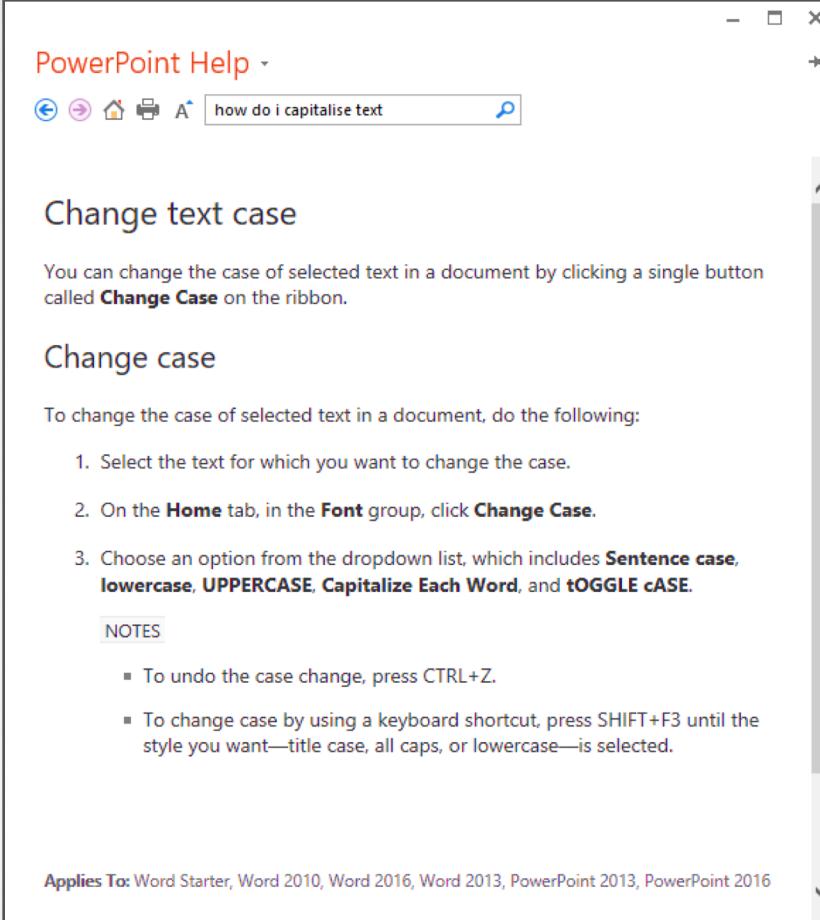
Alternatively, you can visit the [Main Page](#) or read [more information](#) about this type of error.

A project of the [Wikimedia Foundation](#)

10. HELP AND DOCUMENTATION

- 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.

10. HELP AND DOCUMENTATION



The screenshot shows a Microsoft Windows-style help window titled "PowerPoint Help". In the search bar at the top, the query "how do i capitalise text" is entered. The main content area displays a section titled "Change text case" with the following text:

You can change the case of selected text in a document by clicking a single button called **Change Case** on the ribbon.

Below this, another section titled "Change case" provides instructions:

To change the case of selected text in a document, do the following:

1. Select the text for which you want to change the case.
2. On the **Home** tab, in the **Font** group, click **Change Case**.
3. Choose an option from the dropdown list, which includes **Sentence case**, **lowercase**, **UPPERCASE**, **Capitalize Each Word**, and **tOGGLE cASE**.

NOTES

- To undo the case change, press **CTRL+Z**.
- To change case by using a keyboard shortcut, press **SHIFT+F3** until the style you want—title case, all caps, or lowercase—is selected.

At the bottom of the window, the text "Applies To: Word Starter, Word 2010, Word 2016, Word 2013, PowerPoint 2013, PowerPoint 2016" is visible.

Heuristic Evaluation

- Briefing session with experts
 - Tell your experts what to do. Explain the system, who are the target users, relevant factors such as training. May give experts tasks that are considered key for the users to be able to achieve. [briefing might be face-to-face, via email etc]

Heuristic Evaluation

Evaluation period:

- Experts typically spend 1 – 2 hours alone, independently inspecting the system.
- Probably go through the whole system twice, to understand it all.
- Every time they encounter a problem, they note it down on a form, sometimes a scribe is provided to take notes for them.

Heuristic Evaluation

Debriefing session:

- All the experts come together, go through all the problems they have found, agree a final set of problems and a set of ratings of the severity of each problem:

Usability catastrophe – something that must be fixed before a system is released

Major problem – important to fix

Minor problem – fixing is low priority

Cosmetic problem – only if time allows

Heuristic Evaluation

Collaborative Heuristic Evaluation

- Experts rate the problems as they find them, take the average rating (may drop the group discussion, which is very useful)
- CHE method - experts work as a group, but rate the problems privately, if they think a proposed problem is not a problem, they give it a rating of 0

Heuristic Evaluation

Strengths

- Relatively easy to do, with practice people get good at spotting problems
- No one expert will spot all the problems, hence the insurance of 3 - 5 experts
- The severities create a priority list for the developers to work on to improve a system
- Should pick up major problems

Heuristic Evaluation

Weaknesses

- There are always problems that slip through the net
- Regard it as a first pass of evaluation, get rid of as many problems as you can
- User-based evaluation can then focus on the rest