User-Centred Design

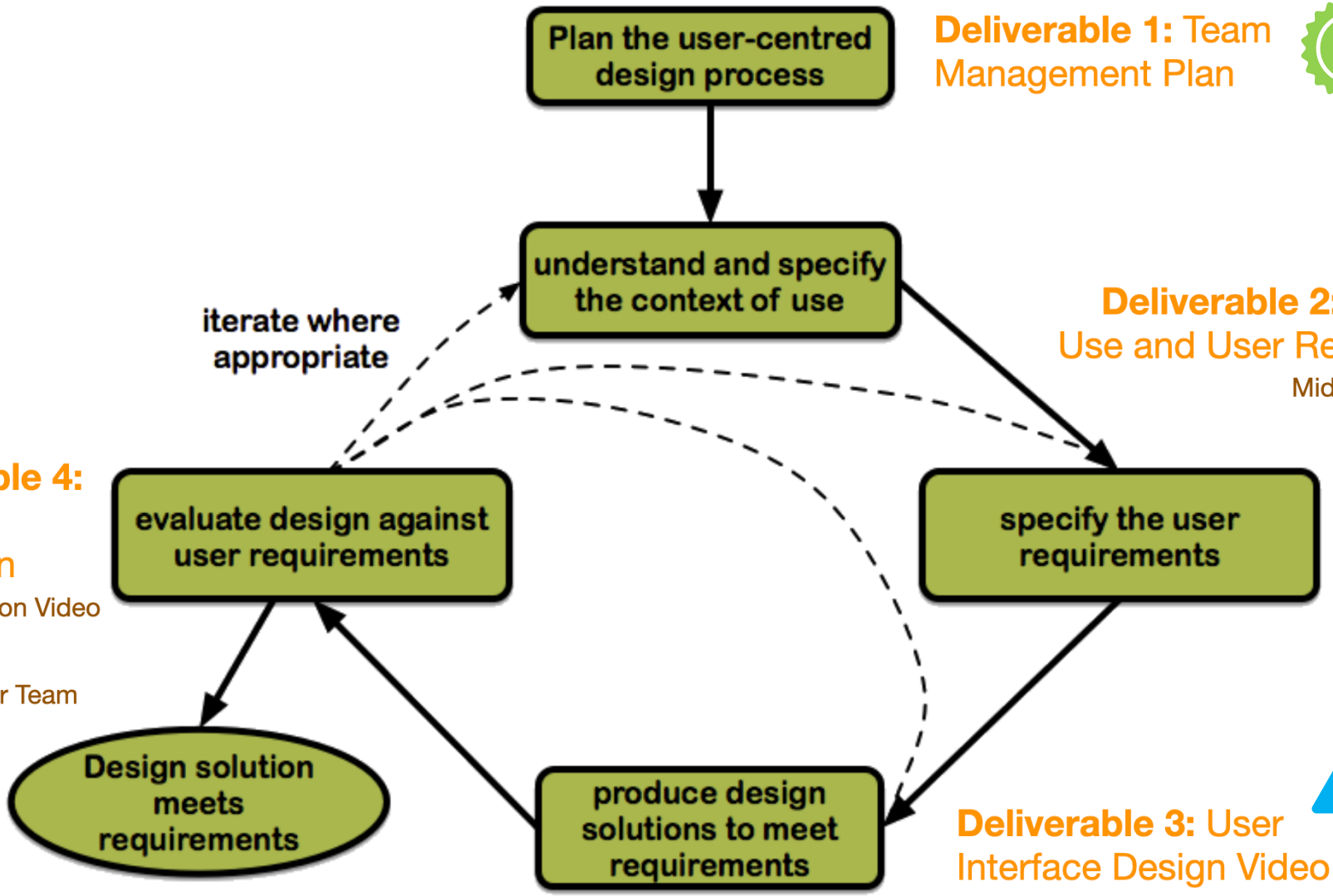
Tutorial 7: Heuristic evaluation

Agenda

- Questions since last week
- Team stand-ups
- User Interface Design
  - Get feedback on your prototype
  - Give feedback on another team's prototype

Depending on progress on prototypes we may do this content in Week 8

UCD Process



Deliverable 1: Team Management Plan

Deliverable 2: Context of Use and User Requirements

Mid-semester Team Reflection

Deliverable 3: User Interface Design Video

Deliverable 4: Usability Evaluation

User Evaluation Video

Ethics Quiz

End-semester Team Reflection

Bring interactive prototype class in Week 7 for bonus marks

SparkPlus

RPF will be applied to UCD 2 results

Mark may be higher or lower than team mark

Individual mark = team mark x individual RPF

For example:

Team mark = 80%

RPF (Mary) = .9 (under average)
Mary's mark = 80*.9 = 72%

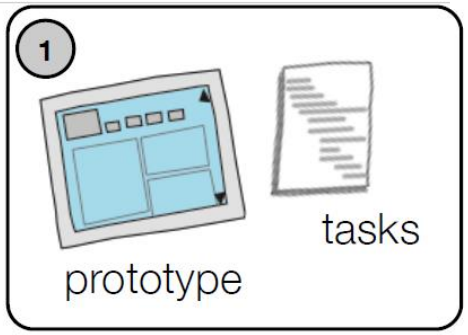
RPF (Jane) = 1.05 (just above average)
Jane's mark = 80*.1.05 = 84%

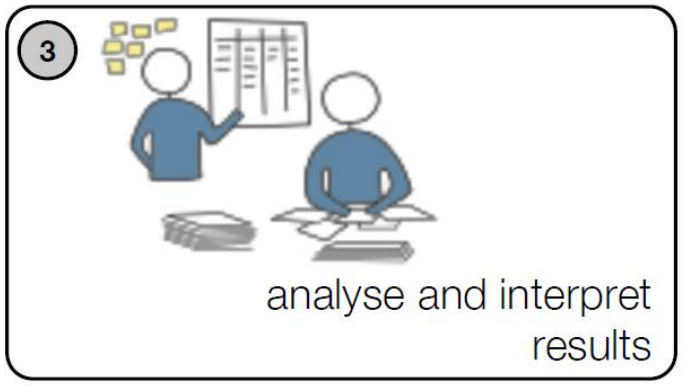
Mark capped at 100%

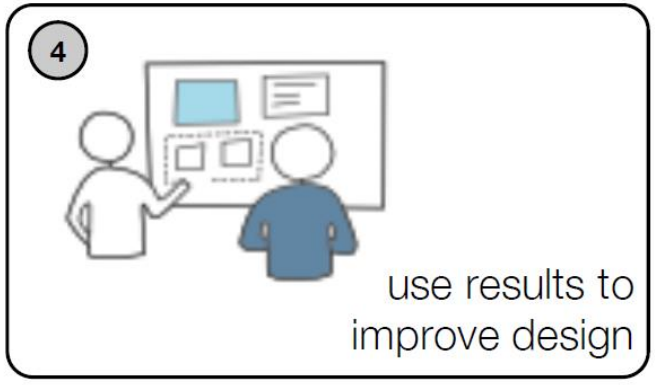
If you don't agree with your RPF, use SparkPlus to raise an objection (available for 3 days after release of SparkPlus result)

SWINBURNE UNIVERSITY OF TECHNOLOGY

Usability Evaluation



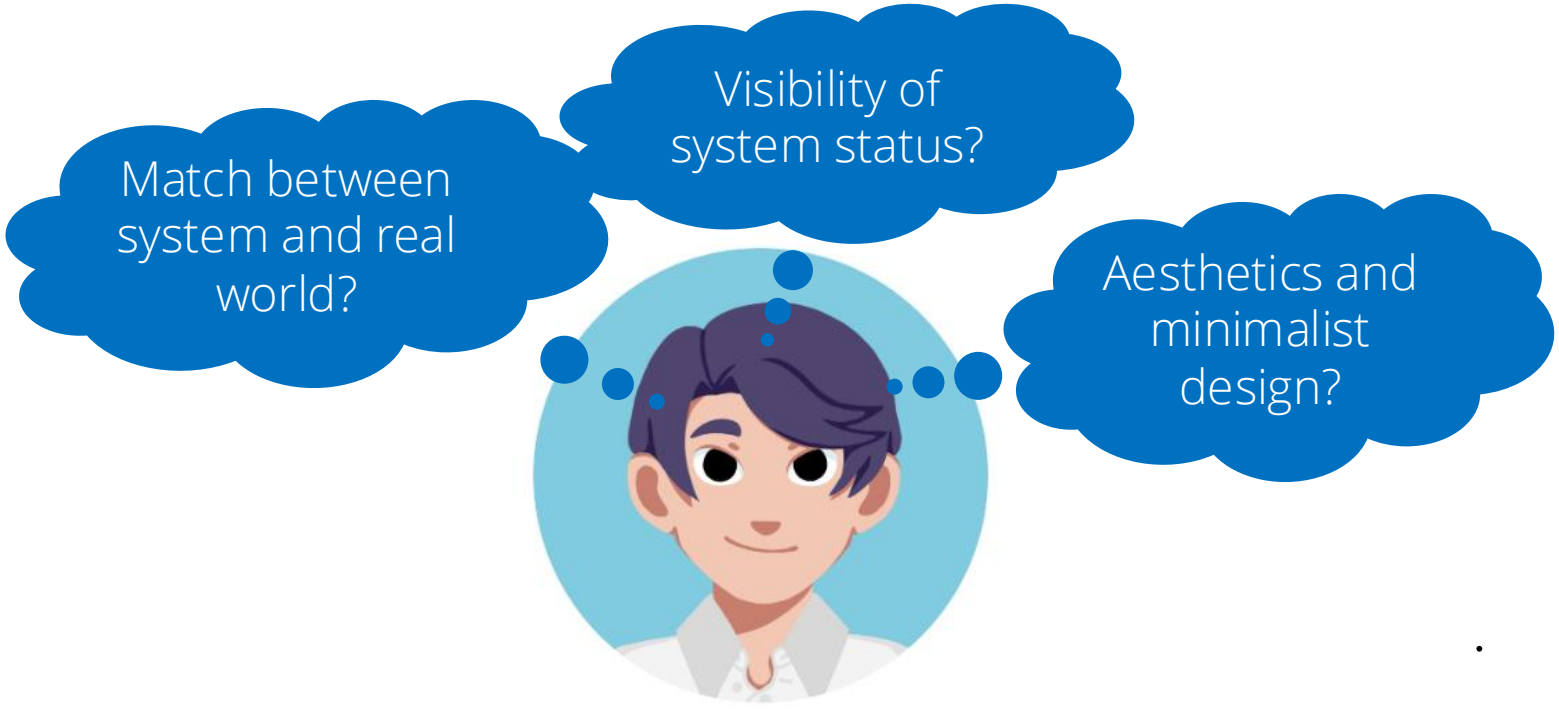




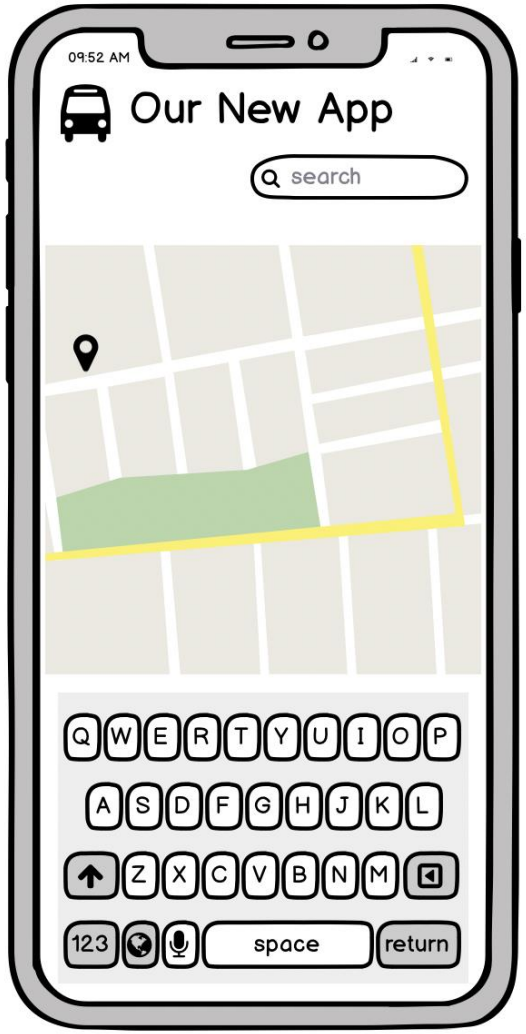
Many ways of doing usability evaluations

Heuristic Evaluation

Usability expert inspects/analyses user interface to identify usability problems.



Usability Expert

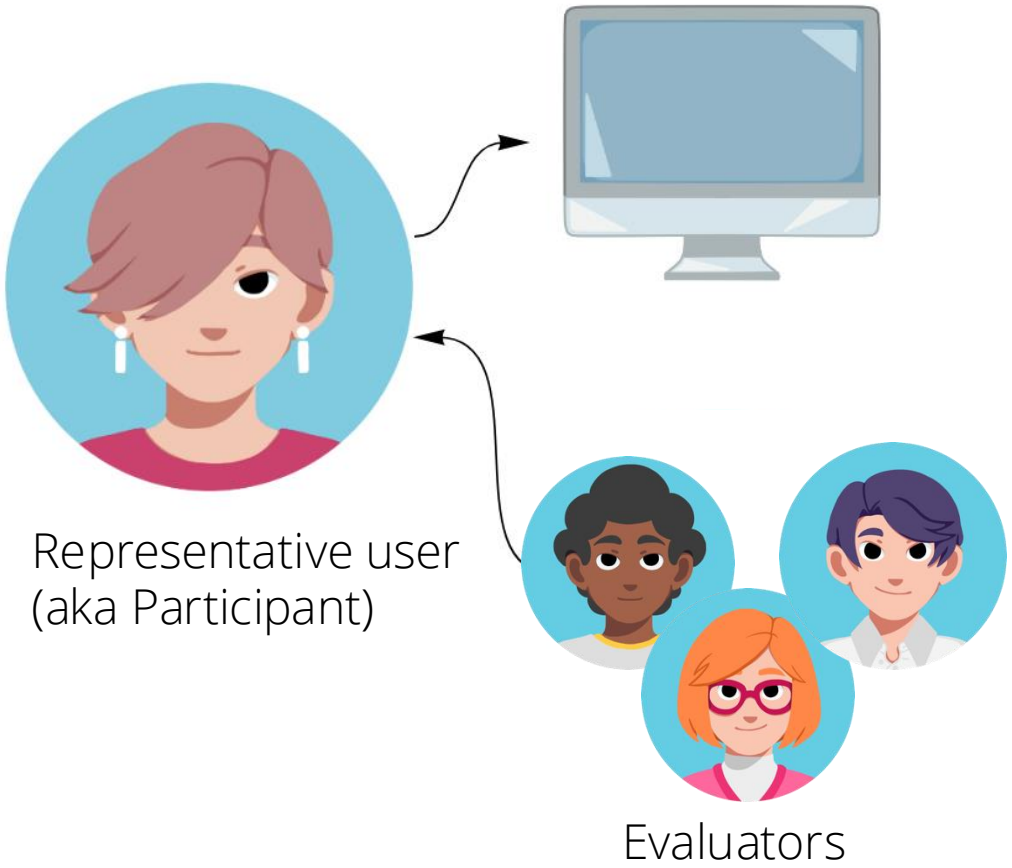


Usability Evaluation Methods

Usability Expert

Non-user methods

- Week 7/8 Heuristic evaluation



Representative user
(aka Participant)

Evaluators

User methods

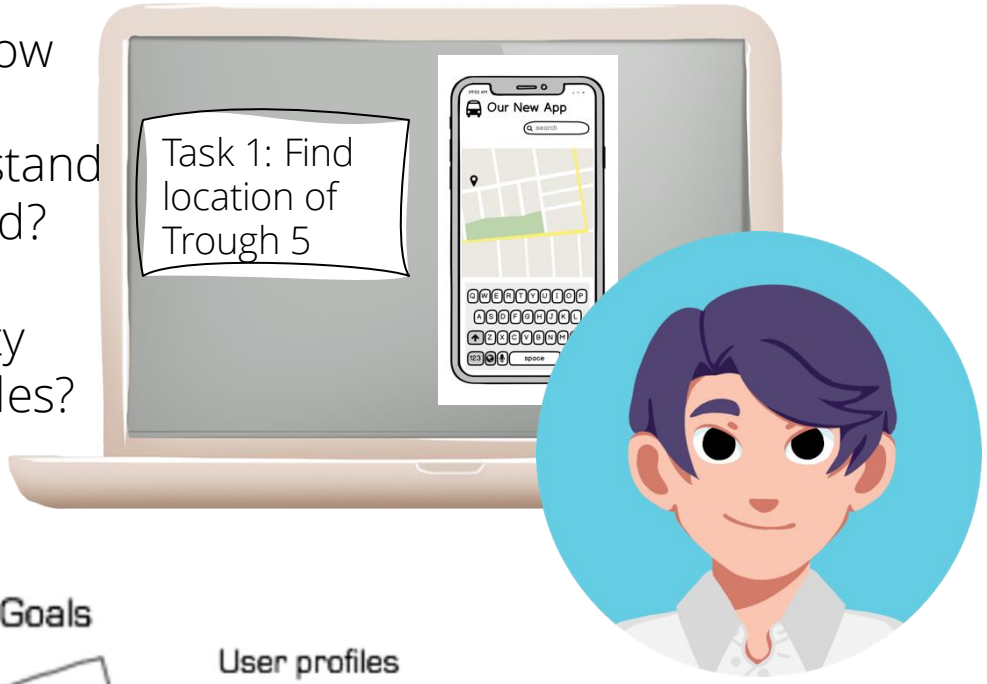
- Week 10 User evaluation

Heuristic Evaluation

Designer
Team

- Ideally just observe evaluation and **take notes**
- Don't give too much away about how your prototype works, let the reviewers explore it…
- If you spend a lot of time explaining a feature, this may indicate you have a usability problem!

- Would the user know what to do next?
- Would they understand what just happened?
- Does the interface conform to usability guidelines/ principles?



Evaluator Team

[Helpful if reviewer thinks aloud in activity]

Heuristic Evaluation - Severity Ratings

Do we really need to fix this issue?

Evaluate severity of problems identified

frequency: common vs rare

impact: easy vs difficult to overcome

persistence: one time problem vs ongoing problem

Rate severity of problem

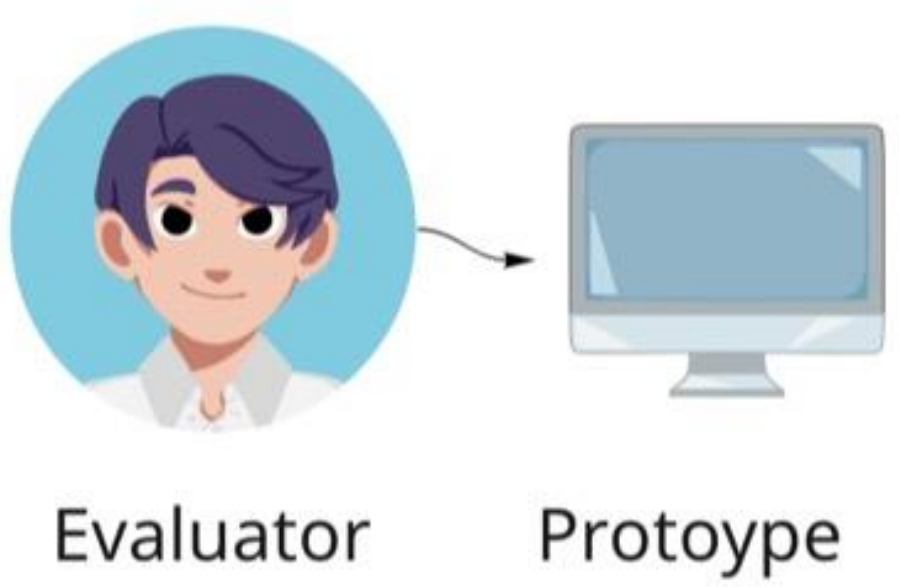
0: No problem

1: Cosmetic problem only: Need not be fixed unless extra time available

2: Minor problem: Low priority fix

3: Major problem: Important to fix, priority should be given to fixing problem

4: Usability Catastrophe: Must fix before releasing product



Heuristic Evaluation

What are the advantages of heuristic evaluations?

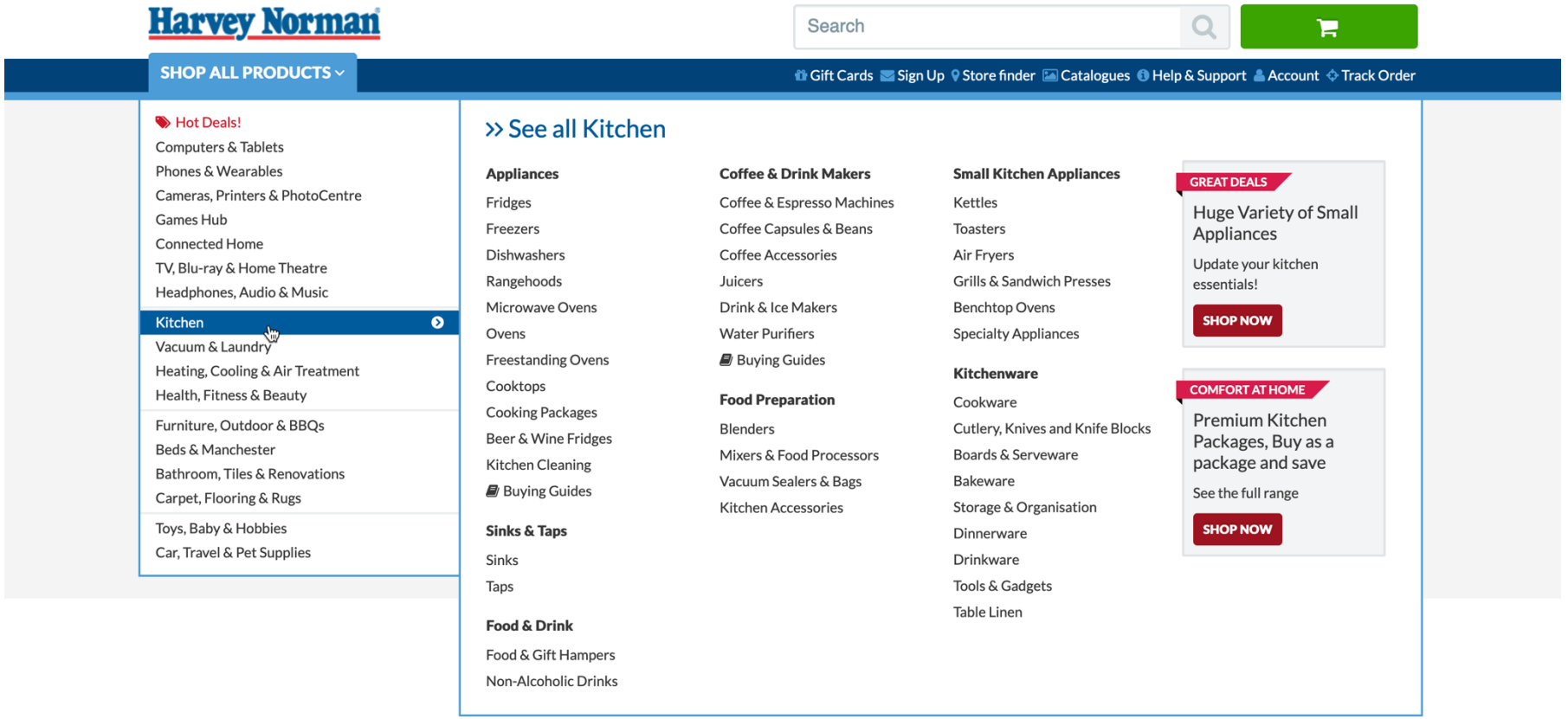
What are the disadvantages?


RESOURCE BOOKER
Now Taking Bookings
If you are new to Resource Booker or require assistance, click here to review some helpful how-to documents. Happy booking!
LOG IN

Design Principles

Graphic design principles: Identify examples of…

- Contrast
- Repetition
  - Alignment
- Proximity

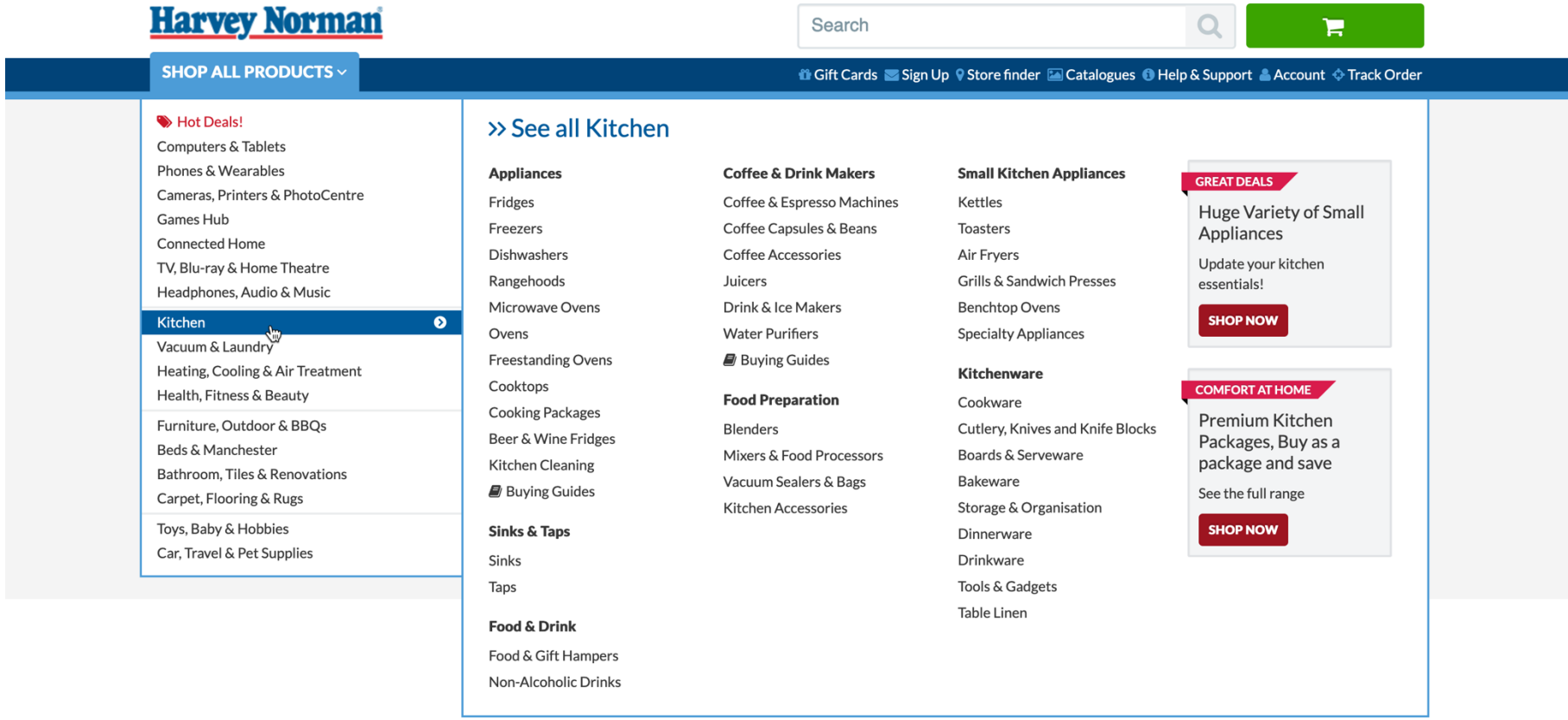


SWINBURNE UNIVERSITY OF TECHNOLOGY

Design Principles

Usability design principles: Identify examples of…

- Visibility
- Feedback
- Structure
- Consistency



SWINBURNE UNIVERSITY OF TECHNOLOGY

Before next week

Review findings of heuristic evaluation (if done this week)

Prep for Evaluation (see Week 8 teaching materials)

Make changes to prototype if required, or prepare evaluation materials

Watch Heuristic Evaluation videos (if you haven't already)

UCD 3: User Interface Video (due W9)

Prepare slides and script and test video recording software

UCD 4: User Evaluation

Watch videos on user evaluation method

Book Participant (a friend/family) to do your user evaluation in Week 10

Nielsen’s Heuristics

1. Visibility of system status

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

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.’

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”.

4. Consistency and standards

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

Nielsen's Heuristics

5. Error prevention

Even better than good error messages is a careful design which prevents a problem from occurring in the first place.

6. Help users recognize, diagnose, and recover from errors

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

7. Recognition rather than recall

Make 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.

8. 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.

Nielsen’s Heuristics

9. 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.

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.