Lecture 2

Usability
Design Walkthroughs
Heuristic Evaluations

UNIVERSITY OF AUCKLAND

SOFTENG 350

Prof. Robert Amor

Usability

Notes from

The UX Book Chapter 21

Heim Chapter 8 and

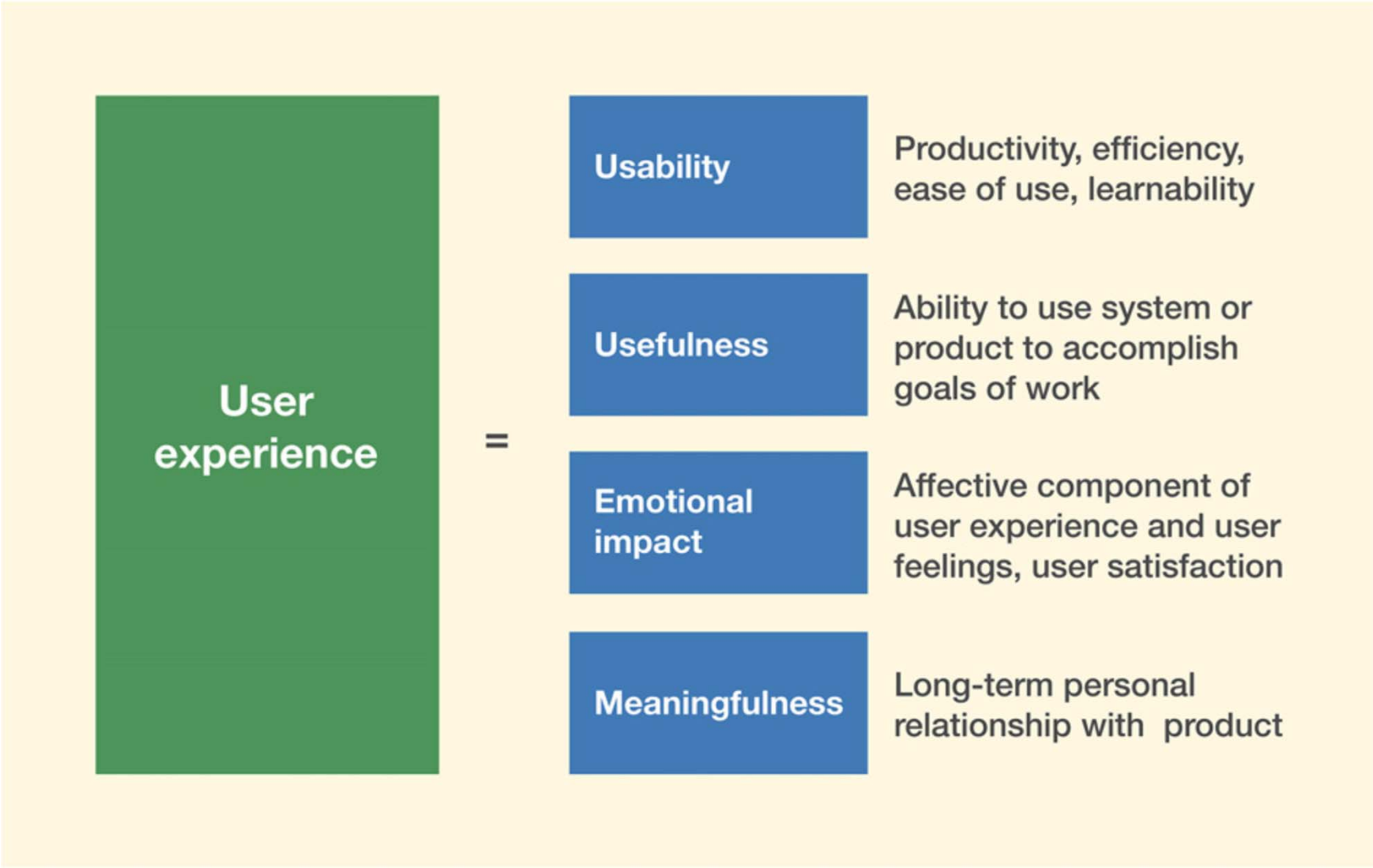
http://www.usability.gov/

Mind Controlled Games

Learning objectives

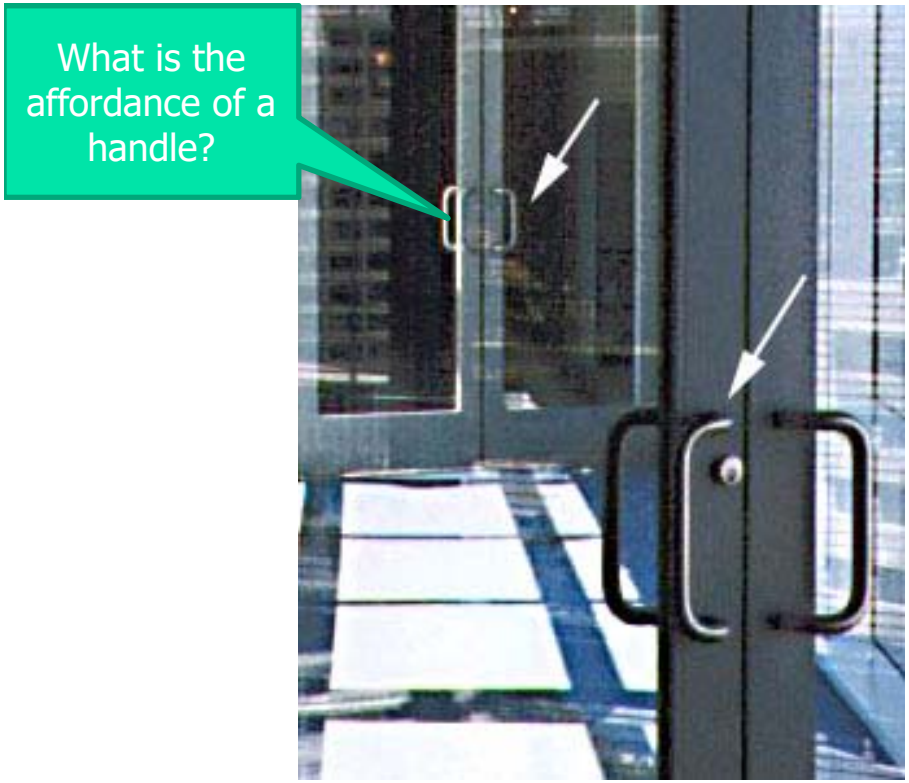
- Be able to define usability
- Be able to describe aspects and measures of usability
- Appreciate that usability depends on context
- Be able to compare and contrast the different usability evaluation techniques
- Be able to select an appropriate usability evaluation technique for a problem

Remember the UX scope



What is the point?

- To ensure that users can use the system as intended!
- There are many examples of bad usability.
- Often problems can be very easily solved!



Not so long ago... (2016)

More recently and local

Thousands of disgruntled customers have complained about Sky Television's "dumb" new software upgrade.

Angry and frustrated clients have taken to the network's Facebook page to condemn the update as slow and causing the picture to lag. **They say it has a small, difficult-to-read font.**

One man's post garnered 16,000 "likes" within hours. Most of the more than 2500 comments written below his post agreed with him.

Stewart Bryan wrote: **"Your new upgrade to MySky is awful - a real backwards step. Smaller font I can't read, terrible colours and so so SLOW to react"**

He said it "might be time to look at Netflix or others".

Other people began re-posting his complaint to Sky, while many more offered their own complaints.

http://m.nzherald.co.nz/business/news/article.cfm?c_id=3&objectid=11573406

The first time you user test your own software

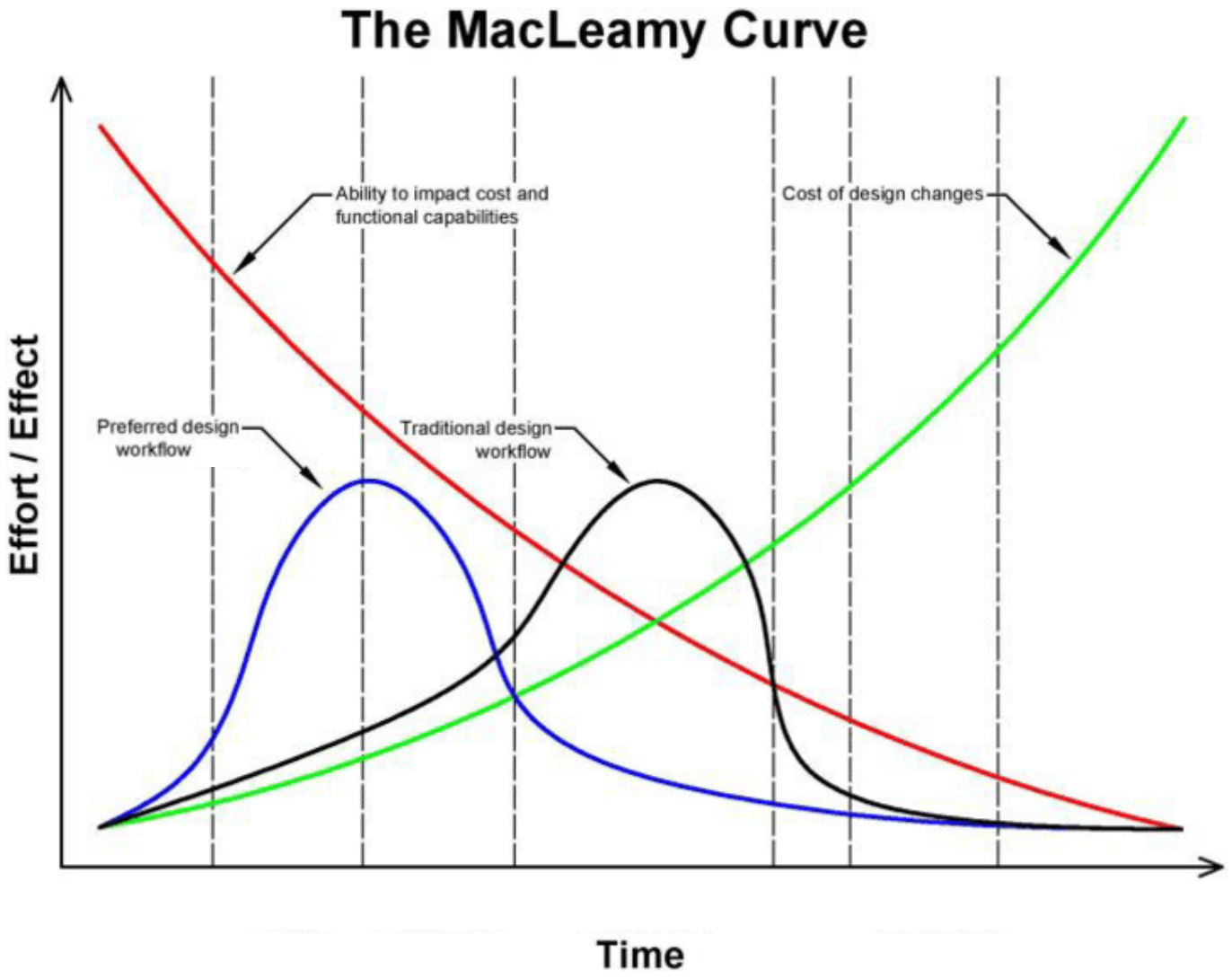
1. You will be horrified at how bad it is!

2. You will find most of the problems are easy to fix

3. You will become intolerant of poor usability!

(Well, maybe that's not all true, but that's the general direction…)

Engineering heuristic



What is usability?

- Usability is the measure of the quality of a user's experience when interacting with a product or system (www.usability.gov 2006)
- Usability is a quality attribute that assesses how easy user interfaces are to use (Nielsen 2003)
- ISO 9241-11, 1997 (Usability: Definitions and concepts)
  - Ease of use
  - User performance and productivity
  - Efficiency
  - Error avoidance
  - Learnability
  - Retainability (ease of remembering)

Usability Factors

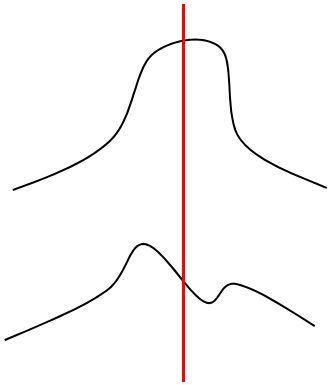
- **Fit for use** (or functionality) – Can the system support the tasks that the user wants to perform
- **Ease of learning** - How fast can a user who has never seen the user interface before learn it sufficiently well to accomplish basic tasks?
- **Efficiency of use** - Once an experienced user has learned to use the system, how fast can he or she accomplish tasks?
- **Memorability** - If a user has used the system before, can he or she remember enough to use it effectively the next time or does the user have to start over again learning everything?
- **Error frequency and severity** - How often do users make errors while using the system, how serious are these errors, and how do users recover from these errors?
- **Subjective satisfaction** - How much does the user *like* using the system?

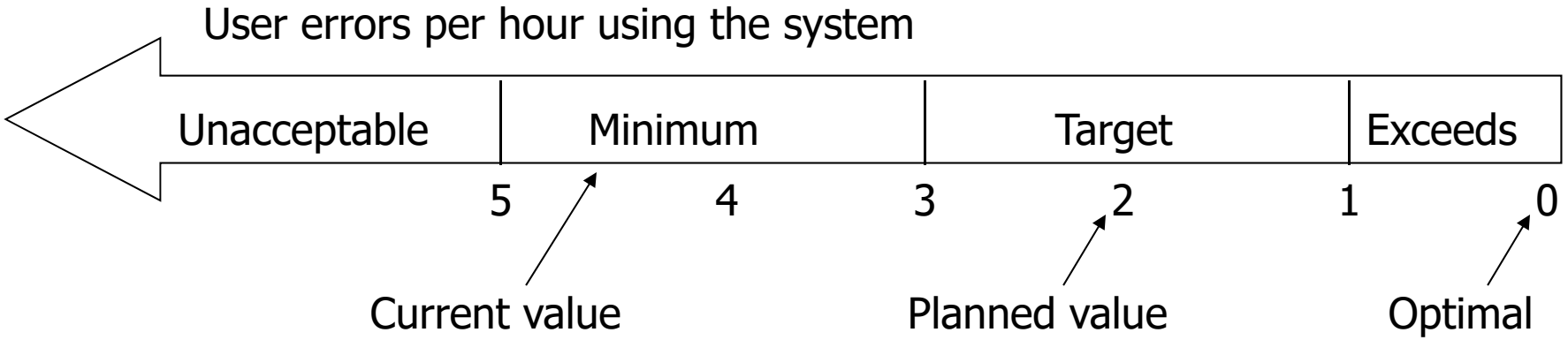
Fit for use

- Does the system function as expected
  - Do the users meet their goals in a timely fashion?
- Finding 'bugs' - otherwise know as errors is NOT the goal!
  - Usability testing ≠ System testing
  - But you are aiming to elicit any usability problems if they're there
- Keep in mind: What are the system goals? E.g.
  - To achieve a specific state
    - Book a flight
    - Pay someone the correct amount
  - To participate in a computer mediated experience
    - Play games

Fit for use measures

- Not necessarily a 1-dimensional measure; may have multiple threshold requirements
  - Average = 2 errors /hour
  - Over 50% less than 1 /hour
  - Less than 5% over 5 /hour
- Can have varying levels of success
  - E.g. minimum: not worse than the old way!





Ease of learning

- Do you expect to have to read a manual or the help?
  - Or an online tutorial, or take a training class, or use the help hotline?
- How much time are you prepared to invest in
  - Learning a new interface?
  - Finding something on a web site?
- What are your usability expectations for a
  - programming IDE?
  - a mobile phone?

Consider

- Identify when you last read the help screens for an app you installed on your mobile
  - What induced you to use the help screens?
  - Did you learn how to make the app more usable?

Efficiency of use

- Do you use the self checkout at the supermarket?
- Do you use ATM machines?
- Is it quicker to order a pizza over the phone or on-line?
- Which of these is cheaper for the retailer?

- How about business processes?
  - If every academic member of staff at UoA (~3,000) *wastes 1* hour a year because the annual performance review system has a poor interface

    = 3,000/1,700(work hrs per year)

    = 1.76 work years lost productivity

Mouse vs Finger

	Mouse	Fingers
Precision	High	Low ("fat-finger problem")
Number of points specified	1	usually 1 2–3 with multi-touch
Number of controls	3: left/right button, scroll wheel	1
Homing time?	Yes	No
Signal states	Hover, mouse-down, mouse-up	Finger-down, finger-up
Accelerated movements	Yes	No
Suitable for use with huge screens (30-inch or more)	Yes, because of acceleration	No: arm fatigue
Visible pointer/cursor	Yes	No
Obscures view of screen	No, thus allowing for continuous visual feedback	Yes
Suitable for mobile	No	Yes: nothing extra to carry around
Ease of learning	Fairly easy	Virtually no learning time
Direct engagement with screen and "fun" to use	No: an indirect pointing device	Yes
Accessibility support	Yes	No

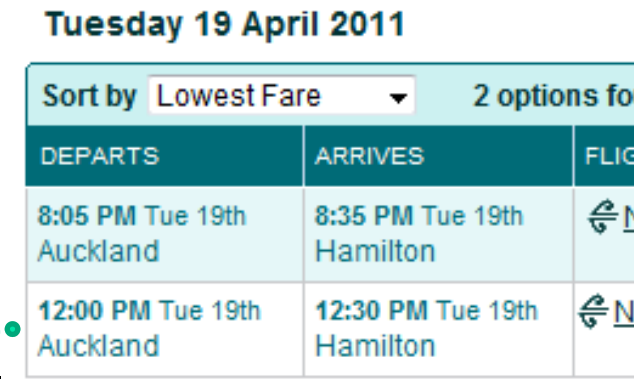
https://www.nngroup.com/articles/mouse-vs-fingers-input-device/

Memorability

- Are there some things you just can't remember how to do?
- A good interface is one where you remember or can, with prompts, recall what to do
- 'Good enough' is contextual
  - If most users are expected to use a feature often, they shouldn't have much trouble remembering how to use it
    - Catering for infrequent users puts on the pressure
  - E.g., for academics, functions related to end-of-semester marks can be a problem
    - Use max of twice a year (so we are infrequent users)
    - Only get to use the feature a handful of times before the system changes!
    - Its high stakes – you guys want your grades to be right

Error frequency and severity

- How frequently do users make errors?
  - Touch phone typing ☹
  - Long and complicated forms
    - It might reduce errors to have dropdown lists to choose from
    - But it might be slower than entering with auto complete.
- What is the cost of errors
  - A patient suffers seizures because a drug-drug interaction was missed!
  - Booking a flight for 8pm when you wanted 8am



Human error - slips and mistakes

Slip – I knew how to do that, but I failed

-  understand system and goal
-  correct formulation of action
-  incorrect action

Mistake – I didn't really know what I was doing

-  may not even have right goal!

Fixing things?

slip – redesign interface to make task easier to execute

mistake – need to help the user better understand the system

Subjective satisfaction

- If users *like* the interface
  - They will make less errors
  - They will persist longer when they are having problems
- Aesthetics – how *nice* it looks is incredibly important in this respect
  - There can be a lot underpinning a perception of looking 'nice'
    - Good contrast / font size, clean layout, good grouping
    - Appropriate language
    - Good fit to workflow
  - 'Ugly' may mean fatiguing, inefficient or apt to produce errors

Types of Usability Evaluations

- Performance measurements
- Heuristic evaluations
- Usability studies
- Comparative studies (advanced class COMPSCI 705/ SOFTENG 702)

http://www.usability.gov/methods/

Performance Measurements

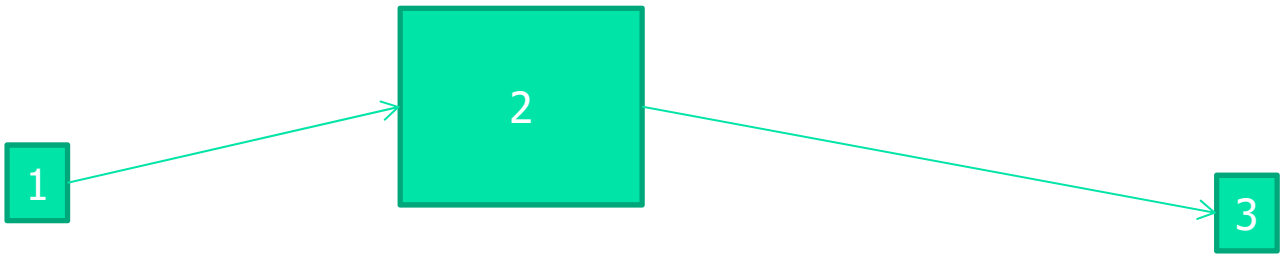
- I.e., an *analytical* performance measurement that can be extracted directly from the interface as compared to an *empirical* performance measurement observed in a usability study
  - Fitts' Law is the classic performance measure for time to complete the task of pointing at an object
  - Hick–Hyman Law time taken to make a decision (e.g., *that* is the object I want!)
- There are other more comprehensive models we won't cover here
  - KLM – keystroke-level model
  - GOMS – goals, operators, methods and selection rules

Fitts' Law

- Fitts' Law is the classic performance measure.
  - Time to target depends on target width (W) and distance to move pointer (D)

$$T = a + b \log_2 \left(1 + \frac{D}{W}\right)$$

  - It is a very valuable measure for designing
    - Control size and location
    - Its also fun to play with!



Hick–Hyman Law

- The time it takes for a person to make a decision as a result of the *n* possible choices

$$T = b \cdot \log_2(n+1)$$

- Particularly important for menus
  - Although $\log_2$ only holds if menu is sorted in a logical order (e.g. alphabetical) – otherwise search time is linear!
- Other factors
  - Recognition time: for icon or word
  - Consistency is good: spatial memory is very powerful –
    - Knowing it's at the left/right side, top/bottom

Usability Evaluation

- Evaluation is an integral part of the development process and can take the form of an informal walkthrough or a more structured heuristic evaluation.
- Formal usability testing can begin once a prototype has been developed.
- Evaluation of an existing system is often part of the analysis stage
- Starting the evaluation process early in the design phase fosters better design because decisions can be tested before it is too expensive to make changes

Multi-choice

With your first prototype of a new app you are concerned about the 'Error frequency and severity' usability factor. A good metric to understand this will be:

a) Productivity loss per hour
b) The distribution of frequency and severity of user errors per hour
c) The total number of mistakes made
d) Average frequency and severity of user errors per hour

Design Walkthroughs
Design Review

Notes from

The UX Book Chapter 25

Channel surfing

Learning objectives

- To be aware of the design walkthrough analytic method for analysis and design phases
- To understand how a design walkthrough/review is undertaken

Common in design disciplines

- For example, Design Critique in Architecture

Design Walkthrough utility

- Informal method to get fast feedback on designs
- Can be used with:
  - Scenarios, storyboards, screen sketches, and wireframes
- Exploring the design on behalf of users
  - Anticipate problems real users might have
- Can be used for early feedback from:
  - Design team, customers, potential users, subject matter experts, and other stakeholders

Design Walkthrough approach

- Use participants, or UX expert, to step through particular functionality
  - Gather reactions to the design for the task

Design Review

- More comprehensive than Design Walkthroughs
- Often used with a click-through wireframe prototype
  - Often a team-based UX inspection
  - Exploring the design on behalf of users

Design Review preparation

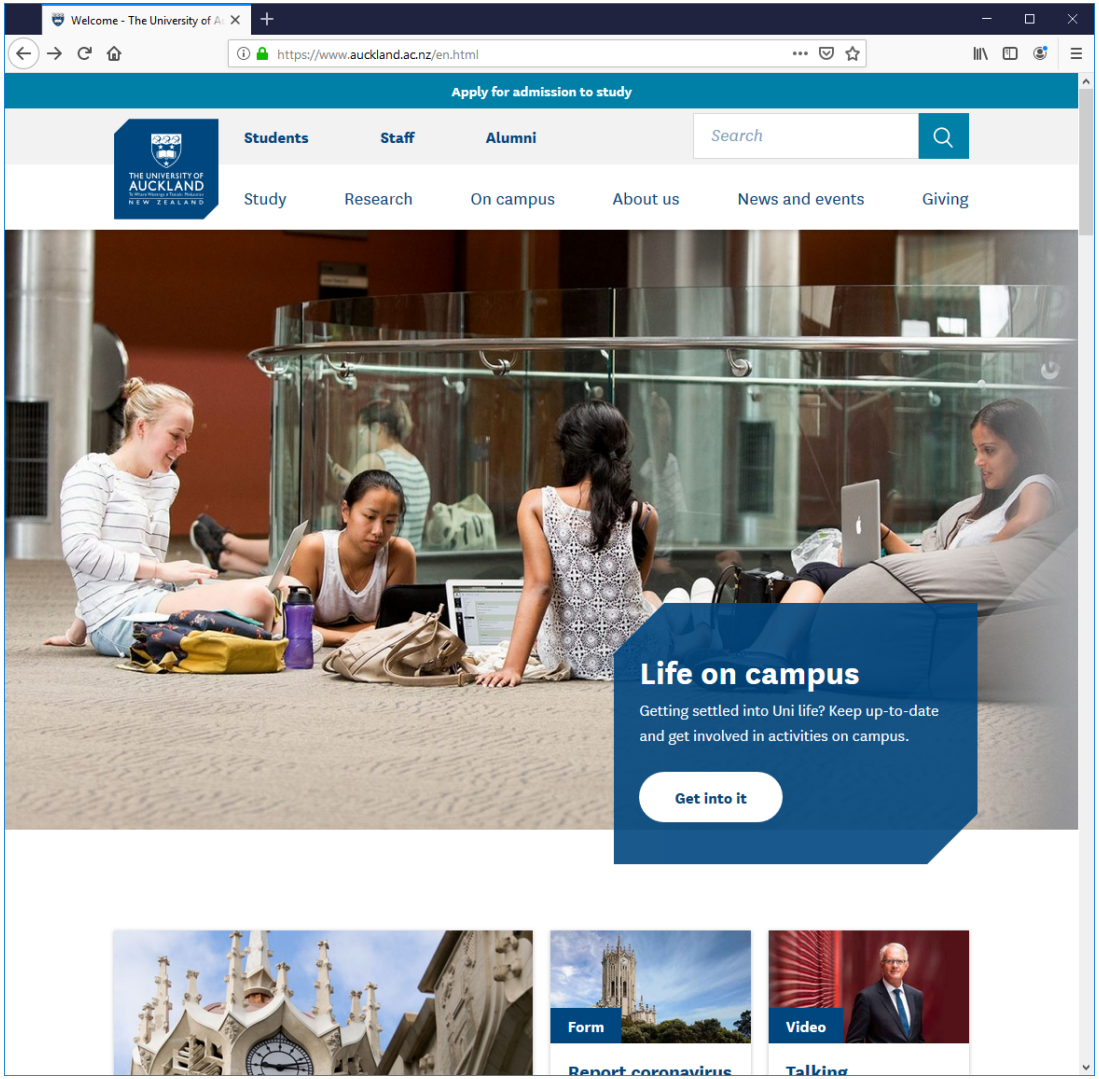
- Start with storyboards for description of the flow
- Have at hand information on relevant users, work roles and user classes
- Practice the design scenarios/stories for the walkthrough
- Schedule a session with users/stakeholders
- Have a leader to run the prototype
- Have a note taker to record UX problems

Design Review session

- Leader introduces design and its purpose
- Leader clicks through the design
  - First for main workflow
  - Then for exceptions, errors, recovery, etc
- Comments invited from the group
- Note taker records UX problems
- Completed within a scheduled time

- Leader explains what user will be doing and thinking, and how the task fits in their work practice

Example: Applying to study

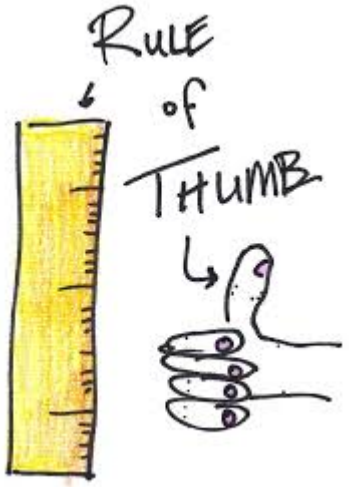


Multi-choice

As user testing gets underway for the university website you discover that many people make a slip at one place in the system. To rectify this you will:

a) Undertake a design walkthrough
b) Design a questionnaire to understand what the users are doing
c) Redesign the interface at that place
d) Provide more help for the user

Heuristic Evaluations



Notes from

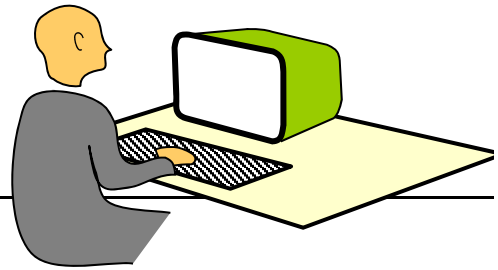
The UX Book Chapters 25 and 26.4

Heim Chapters 5.4-5.6

Learning objectives

- To be aware of a range of heuristic evaluation options appropriate to the analysis and design phase
  - In particular Nielsen's heuristics
- To understand how a heuristic evaluation is undertaken

Heuristic evaluations

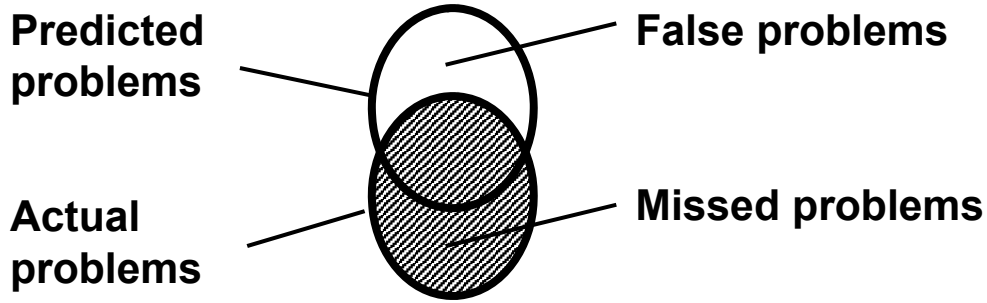


- Expert evaluation
  - An expert looks at a system using common sense and/or guidelines (e.g. Nielsen's Heuristics)

Expert - reviewer

First law of usability:

Heuristic evaluation has only a 50% hit-rate



- More http://www.upassoc.org/upa_publications/jus/2008november/JUS_Kirmani_Nov2008.pdf

Evaluation – *Heuristic Evaluation*

- Heuristic evaluations are performed by usability experts using a predetermined set of criteria designed to measure the usability of a proposed design
- The evaluator follows a scenario through the design and tests each step against the heuristic criteria
- Carrying out a heuristic evaluation is an excellent way to get an understanding of a system that you are going to usability test….
  - But be careful, it can prejudice your study design

- Or a heuristic evaluation can be stand-alone with the evaluator making observations and recommendations based on their experience

Evaluation – *Nielsen's Heuristics*

- In collaboration with Rolf Molich, Jakob Nielsen developed a set of 10 heuristics for interface design.

- The revised set based on an analysis of 249 usability problems.

  http://www.useit.com/papers/heuristic/heuristic_list.html

Nielsen's 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
6. Recognition Rather Than Recall
7. Flexibility and Efficiency of Use
8. Aesthetic and Minimalist Design
9. Help Users to Recognise, Diagnose, and Recover from Errors
10. Help and Documentation

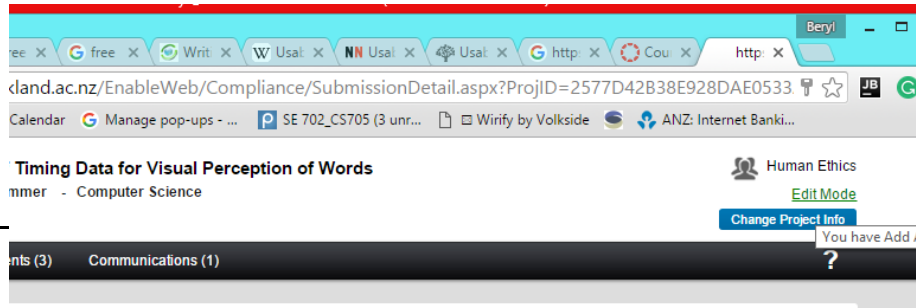
#1 Visibility of System Status

- Keep users informed about what is going on
  - With appropriate feedback, in a timely manner
  - So user feels in control
  - For communication and transparency
  - So user trusts the system

  - Particularly important in transaction tasks
    (e.g., online shopping)
    - Can you tell if there is anything in the shopping cart?
    - Can you tell if you have logged in successfully?
    - Can you tell if your assignment has uploaded?

#2 Match between System and the Real World

- Does the vocabulary match the user's expectations and knowledge?
  - Are you calling the objects on the screen by terms that the user understands (and finds natural)? UPI or User ID?
- Follow real-world conventions
- Does the workflow match the task?
  - Will the user have all the required information at the time you are asking for it?
  - Are they copying from a paper source that lays out the material differently than the data input screen?
  - Are you making them stop in the middle of a task they'd rather not interrupt?
- Is the affordance correct?
  - Is the "Edit Mode" a button




JOHN DEERE
1929 MODEL
KEN FURLONG
JOHN DEERE
1920'S VINTAGE
KEN FURLONG

- **#3 User Control and Freedom**
  - Clearly marked exit
  - Undo / redo
- **#4 Consistency and Standards**
  - Follow platform conventions
  - Terminology stays constant
- **#5 Error Prevention**
  - Remove error-prone conditions
  - Check for errors, get user confirmation

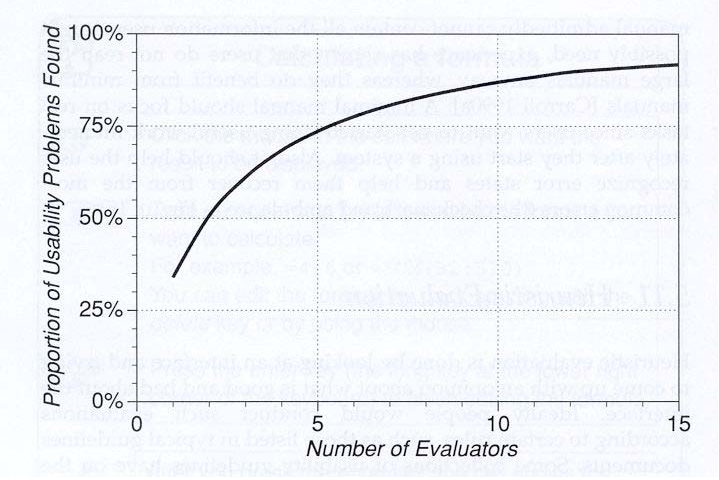
#6 Recognition Rather Than Recall

- Minimise memory load, make actions and options visible
  - If I can put the item on a dropdown list, then I should
    - Why make them type it in and maybe choose an option that's not available?
  - Show the user *something*
    - Maybe you'll get lucky and it'll be just what they want!
  - Basically, use menus and lists instead of relying on blanks

- **#7 Flexibility and Efficiency of Use**
  - Add accelerators for expert users
- **#8 Aesthetic and Minimalist Design**
  - Remove irrelevant, or rarely needed, information
- **#9 Help Users to Recognise, Diagnose, and Recover from Errors**
  - Errors messages in plain language, suggest a solution
- **#10 Help and Documentation**
  - Easy to search, focussed on the task, details concrete steps to carry out

Nielsen's Advice for Heuristic Evaluations

- These are intended to counter the problems of expert evaluations
  - Use multiple independent evaluators
  - Use an observer to record evaluator
  - Go through the interface several times
  - Compare interaction against list of heuristics
  - Use heuristics specific to design
  - List heuristic problems and how the heuristic is violated



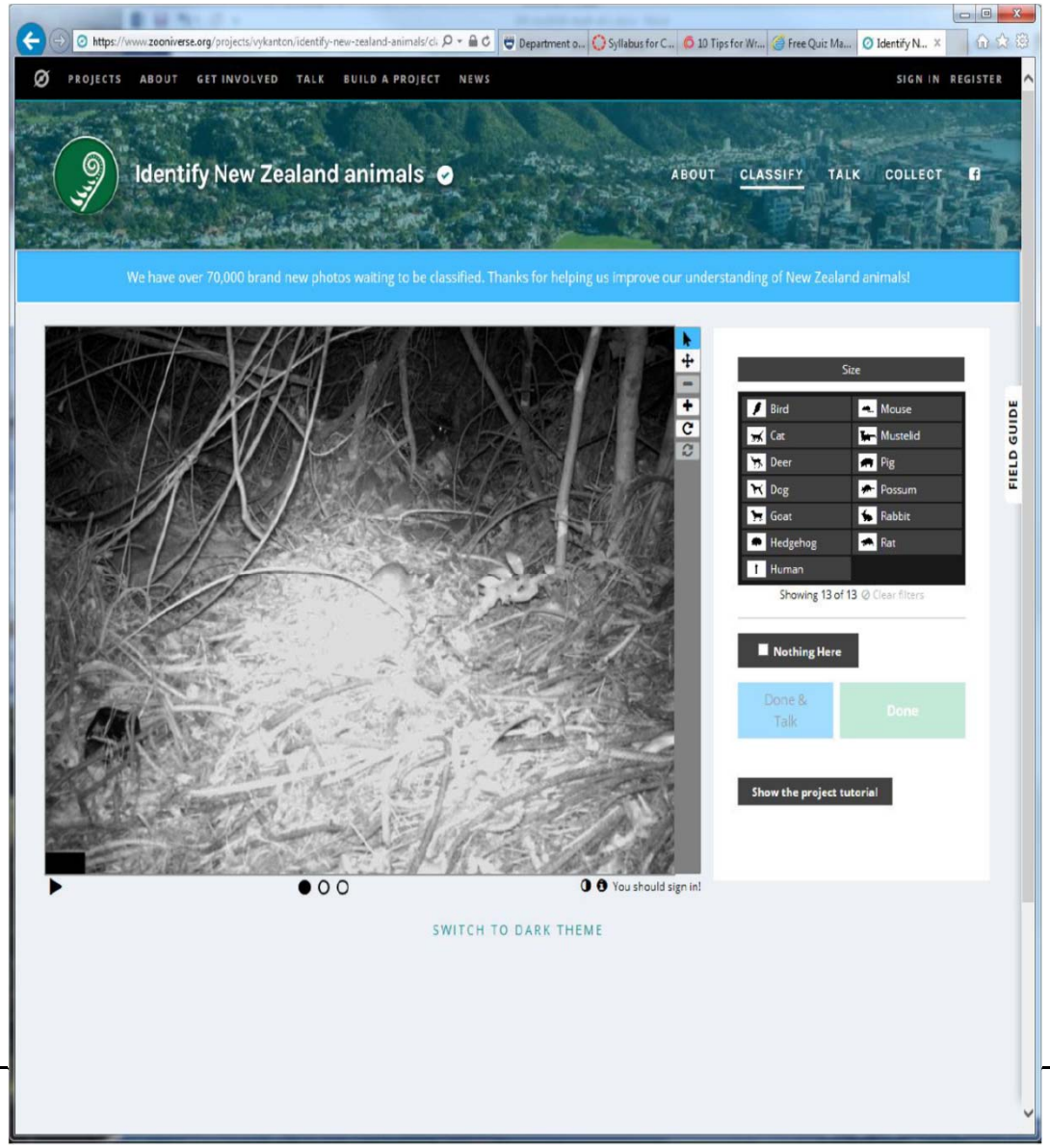
A world of heuristics

- Can be devised for more specific domains
  - For physical format – e.g. web pages
    http://www.psu.edu/webconference/Web2004/Materials/Heuristic.pdf
    - Domain specific concepts like good background graphics
    - 'Housekeeping' like correct spelling & grammar
    - May want to evaluate the search function
  - For task domain – e.g. in health…
    - Is patient name and date-of-birth clearly visible at all times?
    - Does the interaction fit to clinical workflow?
- Can be quite long
  - About 100 heuristics for mobile apps:
    www.tmap.net/sites/default/files/Checklist_Mobile_App_Testing_0.docx

Summary

- Heuristic evaluations are
  - Expert reviews and often include
    - Modelling the interface (e.g., Hierarchical Task Analysis)
    - Evaluating against a set of guidelines

Multi-choice



Multi-choice

The selection box on the right of the screen, used for identifying animals, is a good example of Nielsen's heuristic:

a) Aesthetic and minimalist design
b) Visibility of system status
c) Recognition rather than recall
d) Help and documentation