INFO2222

Computing 2

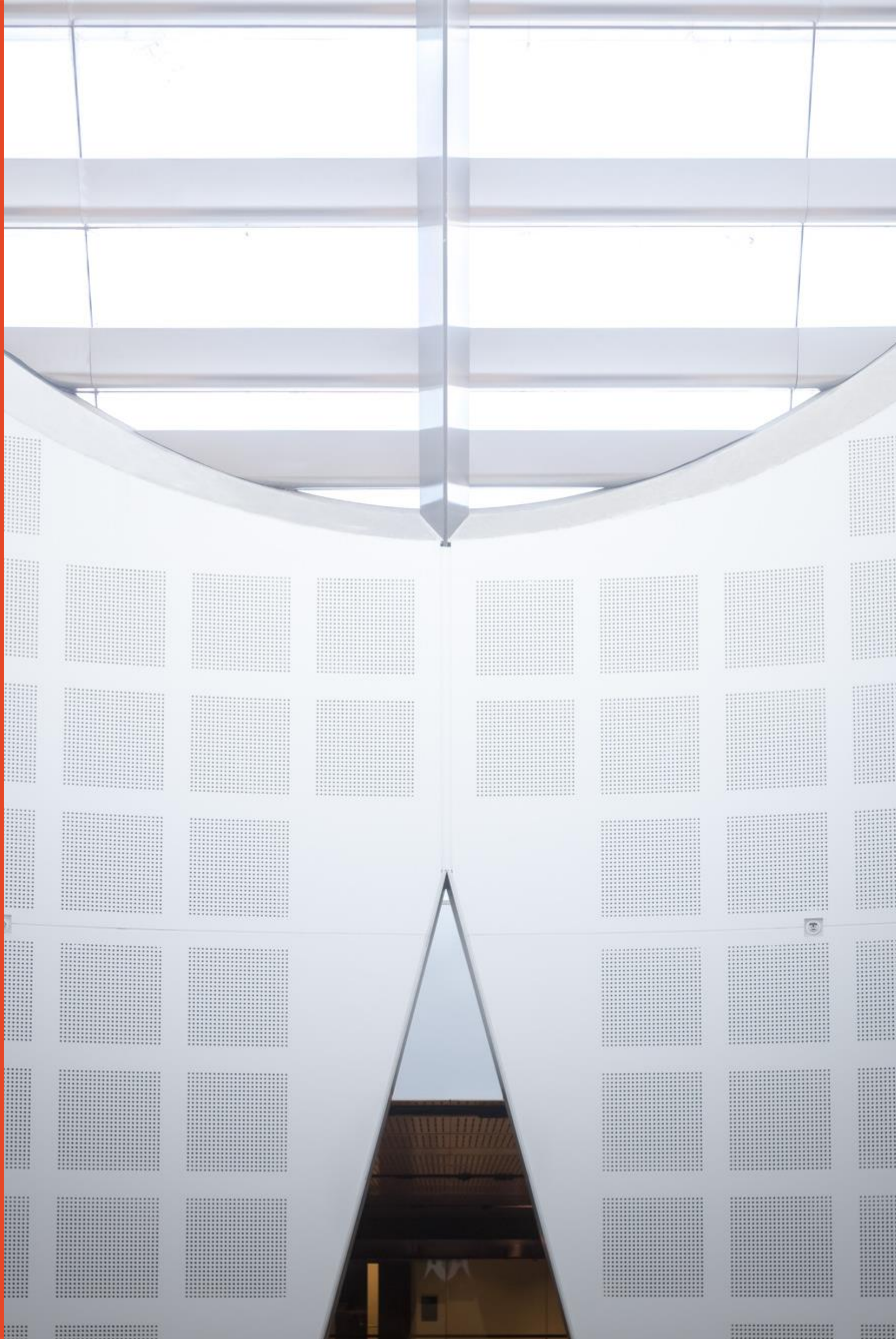
Usability & Security

Lecture 3: Design & Implementation

Presented by

Joshua Burridge

THE UNIVERSITY OF SYDNEY



Design & Implementation – Feedback

- How do you test memorability?
- Why test with only 5 people?
- How do you analyse the qualitative data from think-alouds?
- What is persona? How is it different to characteristics?

Design & Implementation – Feedback

- Soft system vs hard system
- What is Situation of Concern?
- What does Transformation mean?
- Weltanschauung vs Worldview?
- Which stakeholders do we need to consider?
- STS vs CATWOE?
- STS real world example?

Design & Implementation – Recap (Lecture)

- ***Soft Systems*** like usability and interfaces require methods to handle ***stochastic*** processes
- We can't know or predict a single stochastic outcome with certainty – we can know and predict ranges and groups of them
- ***Soft Systems Methodology*** uses this to understand soft systems

Design & Implementation - Recap (Lecture)

CATWOE:

- **Customer/Client**
- **Actors**
- **Transformation**
- **Weltanschauung/Worldview**
- **Owner**
- **Environmental**

STS:

- **Situation of Concern**
- **Transformation**
- **Stakeholders**

Design & Implementation - Recap (Lecture)

- Usability is a relationship between a system/product and its users
- We can view this relationship as one of communication: the designer to the system/product (system image), then the system/product to the user
- Each step along the way involves translation between conceptual/mental models with embedded contexts and potential for miscommunication, leading to misalignments
- Therefore we need ways to evaluate our designs after we create them

Design & Implementation - Recap (Lecture)

Think-aloud:

- A powerful method to gain insight into the user's experience with a system/product
- A facilitator observes a user performing a specific, concrete task
- These observations allow the facilitator to somewhat understand the user's mental models
- Great for cheap, effective, easy evaluation with small numbers of participants

Design & Implementation - Recap (Tutorial)

- You have hopefully seen how individual preparation can make a tutorial more useful for the whole group
- You should also have run through a substantial think-aloud activity and be confident at applying this technique for your assignment
- You should all be in a group – anyone *not* in a group (in Canvas, with access to the group page on Canvas) should see me after the lecture.
- You should all be getting comfortable with Bottle. We will rapidly be moving on to web development from this week, do not let yourself fall behind!

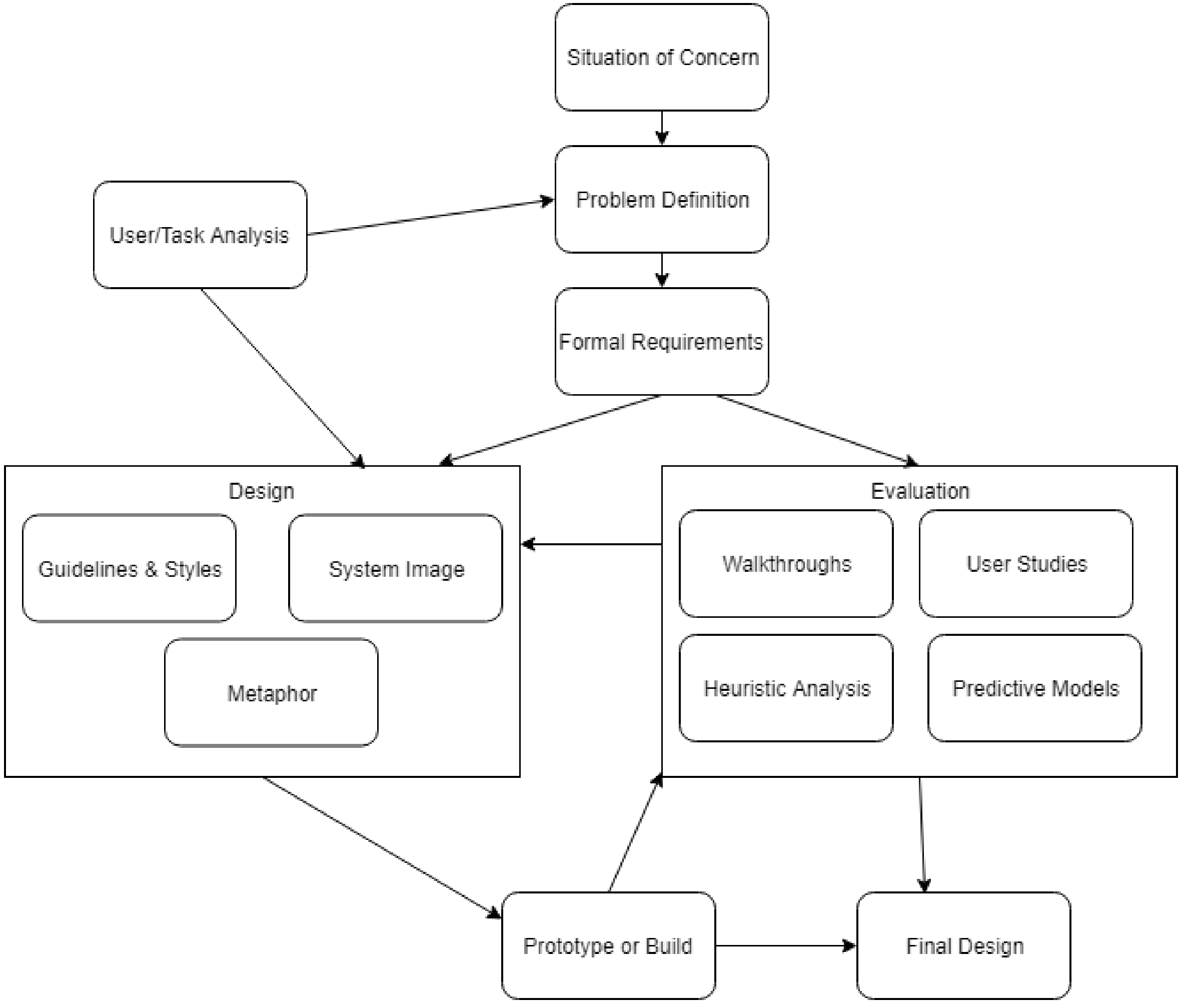
Design & Implementation – Learning Outcomes

- Explain the iterative approach to usability design and implementation
- List methods available for each phase of the iterative approach
- Identify the role of usability design within the wider system development lifecycle (SDLC)

Design & Implementation - Iterations

- We know from earlier weeks that the design phase of usability engineering is inaccurate, and the evaluation phase (if done correctly) is accurate
- We also know that because of this usability engineering is typically done by design, evaluate, redesign, re-evaluate, etc.
- This cycle is known as the ***iterative approach*** (also iterative design, iterative testing, continuous improvement etc.)
- This has become a common theme in a lot of IT/SoftEng/CompSci fields – like Agile, & ITIL – as well as in manufacturing – like Six Sigma & Lean manufacturing

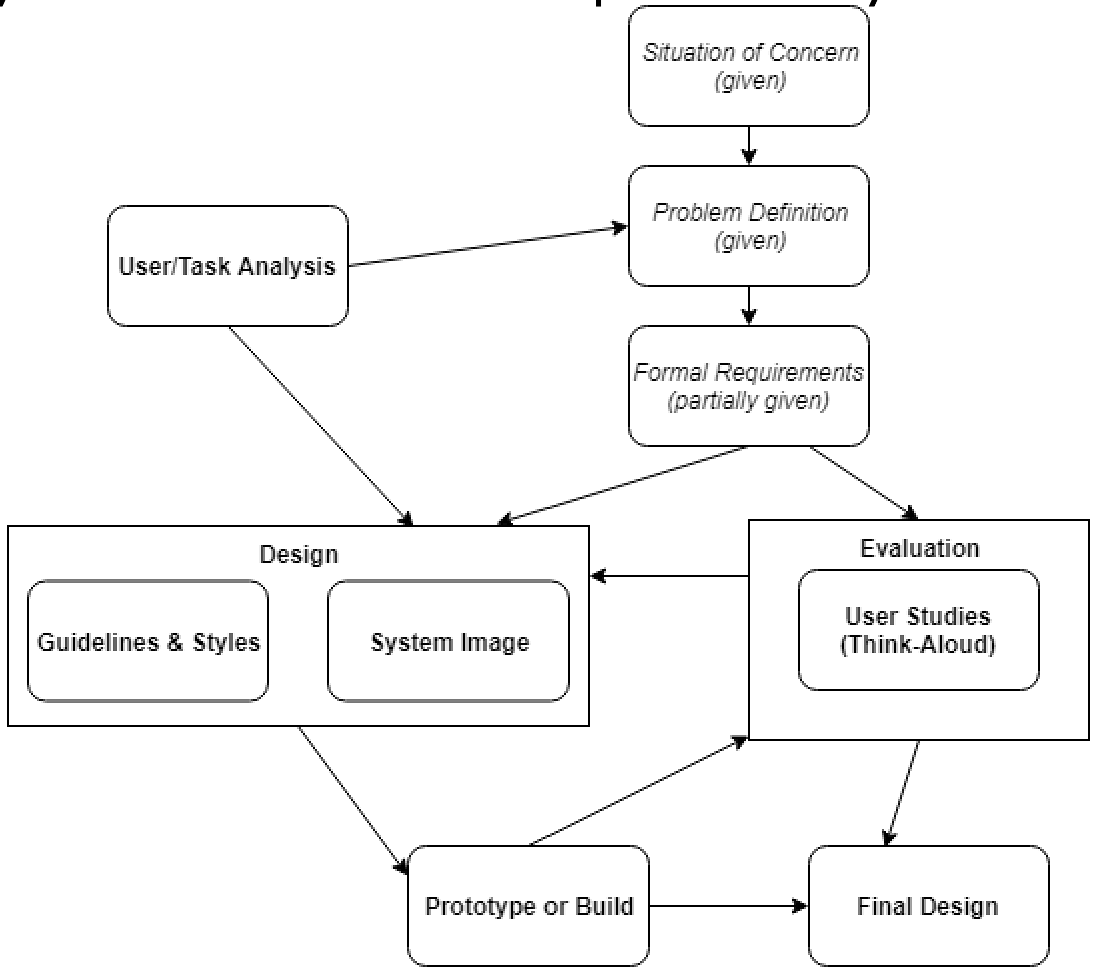
Design & Implementation - Iterations



Reproduced from Bowman, D. Virginia Tech

Design & Implementation - Iterations

– In INFO2222 we apply only a small selection (we cover the theory only of a few more techniques which you will need for the exam)


Situation of Concern (given)
Problem Definition (given)
User/Task Analysis
Formal Requirements (partially given)
Design
Guidelines & Styles
System Image
Evaluation
User Studies (Think-Aloud)
Prototype or Build
Final Design

Design & Implementation - Iterations

- The core phases:
- ***Analysis***
  - What are you trying to achieve? For who? With what?
- ***Design***
  - How are you going to achieve it?
- ***Implementation***
  - Draw something, code something, build something.
- ***Evaluation***
  - Did it work? Have we satisficed the requirements?
- If not, go back to whichever step your evaluation indicates did not achieve its purpose. Did you aim at the wrong users? Was your design flawed? Is the implementation broken?

Design & Implementation - Iterations

- You have enough already to *eventually* create a perfect interface (or as close to perfect as possible)
- Simply keep making interfaces and evaluating them infinite times until you get it right!
- This is of course very inefficient
- Even though analysis/design/implementation are inaccurate, the more accurate we can make them the fewer and cheaper iterations we will need to obtain a good outcome
- The earlier you detect an issue in the process the cheaper it is to resolve

Design & Implementation - Iterations

How do we do this?

- Learn from others (guides, conventions, styles, etc.)
- Pre-empt issues before we implement them
- Use as short an iteration cycle as possible, with as few changes before testing as possible (this is the motivation behind Agile)
- This is the reason we incorporate 'personas' so early:
  - Personas are 'best guesses' about users
  - They are far from perfect - real users are much better than developers pretending they are users
  - But: once you have researched and created them, they are free to use
  - They reduce the number of issues allowed to make it to the evaluation phase

Design & Implementation - Iterations

- Where does all this fit in the wider development cycle? It *is* the wider development cycle
- Usability must be present in all phases of product or system development in order to be effective – it is not an afterthought to be brought in during testing
- It also has all the same rules as ordinary development
  - Tests and success criteria must be determined before implementation
  - Test early and often
  - Get direct user feedback on the outputs/deliverables
  - Don’t be afraid to throw away things that didn’t work

Design & Implementation – Information Architecture

Information Architecture

- Part of the analysis & design phases
- A field which began with early networked computer systems, but exploded with the internet/world wide web
- Is one of those 'loaded terms' in IT jargon. We will use the following definition:

  > *Information architecture is the discipline of organising and structuring data and functions in a system to support usability.*

- It is (hopefully) the reason you find 'Career Services' in the 'Study' category on the University website, as well as the reason YouTube has a 'suggested videos' section where and how it does.

Design & Implementation – Information Architecture

- The core idea of information architecture comes from a theory of how **data** gets turned into ***knowledge***
- ***Information*** is the intermediate step – but how is it different to data?
- Information is by some definitions data which has been ***organised, processed,*** or ***structured***
- ***Information architecture*** is the choices and designs involved in that organisation, processing, or structuring
- It is a complex field which involves a lot of theory from cognitive psychology, data science, and others – we will be applying the concepts and techniques without going too deeply into where they have come from

Design & Implementation – Information Architecture

- The outputs of information architecture are primarily an organisational ‘scheme’, and an organisational ‘structure’, which when put together lay out the information created from your data
- Organisational schemes are the answer to ‘what data should I put with what data’.
- Organisational structures are the way those groups are (conceptually) stored and accessed (remember we can decouple the reality of the system from its system image)

Design & Implementation – Information Architecture

Schemes come in 3 main types:

- Exact:
  - Examples: Alphabetical, Chronological, Geographical
  - Mutually exclusive categories (1 item can only exist in 1 category)
  - Easy to use if you know what you are looking for, hard to use with partial information, very easy to create (i.e. bad for learnability, good for efficiency)
- Ambiguous:
  - Examples: Topical, Task-oriented, Audience-specific
  - Multiple parents – items can appear in more than 1 category
  - Easy to use if you have only some idea of what you want, hard to find a specific known item, difficult to do well (i.e. bad for efficiency, good for learnability)
- Hybrid - a blend of both, hopefully taking more of the advantages than the disadvantages

Design & Implementation – Information Architecture

- Structures have in the past come in a few forms – ***hierarchy*** (like the file system on a computer, or a family tree), ***database-oriented*** (like the 'advanced search' option in the university library page) and ***hypermedia***.
- Modern structures (especially on the internet) decouple the internal reality of the structure (almost always database-oriented) and the interface version of that structure (almost always hypermedia)
- This then has a totally different conceptual model embedded and overlaid upon it to form the final system image – often involving some flavour of hierarchy, though not always
- This presents new challenges to interface design for usability, but also gives us the tools to do so properly

Design & Implementation – Information Architecture

- ***Hypermedia*** is an idea which has grown out of the ***hypertext*** concept from the original WWW (and earlier, more on this in the case study)
- It is the idea that a user can be taken to different views based not on a separate organisational structure, but by the source content they are looking at immediately
- Like this
- A blindingly obvious idea to those of us who grew up with the web – a revolution in information organisation and access at the time
- The difference between hypermedia and hypertext simply opens it up to video (think of YouTube's UI), audio (think Spotify), etc.

Design & Implementation – Information Architecture

- Hypermedia does not mean you get to ignore your information's structure!
- You need to build that structure in order to support the LMEES of your system
- You should also support LMEES with ***navigational context***
- This is the combination of cues that tell a users 'where am I' and 'where can I go?'
- Navigational context includes tools like a menu bar, cookie crumbs, visual locators, site map, etc.
- It also includes the ***labels*** on your information (categories and items)
- The scheme, structure, and navigational context together should combine with the user's context to form the user's mental model of your system

Design & Implementation – Information Architecture

- We already know the best way to verify we have done the information architecture correctly – think-aloud studies
- But how do we make our design and implementation more accurate? Where do our categories and labels come from?
- We will be using a technique called ***card sorting***
- This is typically a user based study method – though for more efficient iterations you can also use developers acting based on personas.
- There are a few versions, but they all include a series of cards (or just pieces of paper) which each have a single 'item' or 'function' of the system on them
- Users then group these cards to form an emergent structure

Design & Implementation – Information Architecture

Open card sorting

- Participants are given a series of cards, and told to group them
- Participants are then given 1 blank card per group to write a 'category name' on (these are your labels)

Closed card sorting

- Participants are given a series of cards including a series of category cards, and told to group them

Hybrid card sorting

- Participants are given a series of cards including a series of category cards, and told to group them
- They are also told if they feel cards do not belong to any group, to leave them aside and make a new group for them

https://www.usability.gov/how-to-and-tools/methods/card-sorting.html

Design & Implementation – Information Architecture

Things to remember:

- Participants get bored easily. Limit the number of cards (~30 is high, 40 absolute max) and categories, use multiple phases of the study at different hierarchy levels if needed
- Randomise the card order between participants to offset early-sort bias
- Often the study is run with an early open phase with a random selection of items, then a later closed or hybrid phase to check the validity of the output
- Efficiency is important here – you only *need* to sort core, frequently used or important items. Rarely used, unimportant items can be found using search or exact scheme lookup if necessary (your user studies will tell you which is which)

Design & Implementation – Case Study

- How do interfaces go from idea to reality? Let's look at some of the first:
- ***The Mother of All Demos:*** https://www.youtube.com/watch?v=mT_PhstIBIA
- The first time we saw a computer system with:
  - Hypertext
  - A GUI
  - A computer mouse
  - Windows
  - Computer images
  - Video conferencing
  - Word processing
  - Revision control
  - Navigation

Design & Implementation – Case Study

- There are some disputed 'earlier inventions', but none of them came anywhere close to the publicity and impact of tMoaD
- This moved into Xerox PARC, which spawned the technologies which made Apple and Microsoft what they are today
- Much more advanced than what we are doing here of course, but shows the sequence of development for interfaces (and indeed all systems):
  - Someone has an idea
  - That idea gets made into a demo (tMoaD)
  - That demo gets refined into further demos (Xerox PARC)
  - The final product is then released 'into the wild'
- What we're interested in now are the 2 middle steps

Design & Implementation – Prototyping

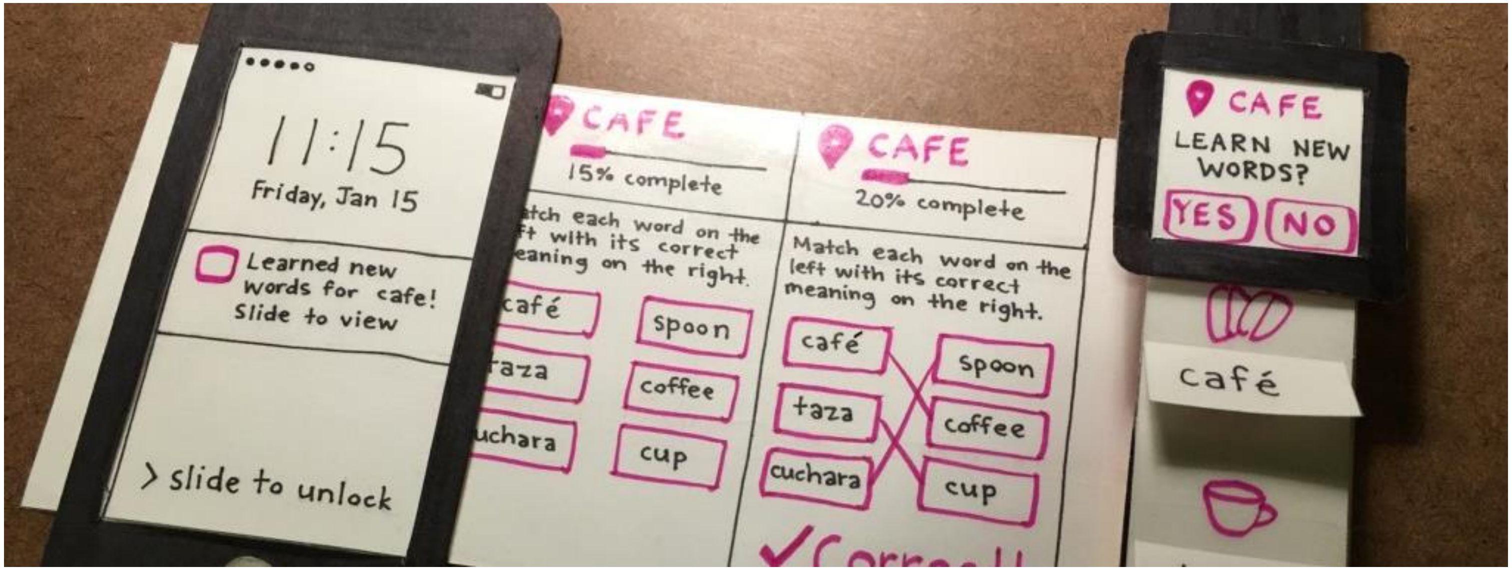
- In order to have something for users to test (e.g. with think-aloud studies) you need to build something
- But remember – the earlier you detect issues in the development cycle, the faster and cheaper it is to fix them
- Because of this, we don't want to wait until our product is complete before we test it!
- We need to 'sort of' build our product – to produce something that is going to represent the finished thing, but not cost as much, or not need as much of the design to be finished, or be faster to modify
- We need a ***prototype***

Design & Implementation – Prototyping

- Prototypes are representations of your intended finished product
- They have many different levels of completeness and sophistication
- Their goal is to allow testing as early as possible, as cheaply as possible, and with as valid results as possible

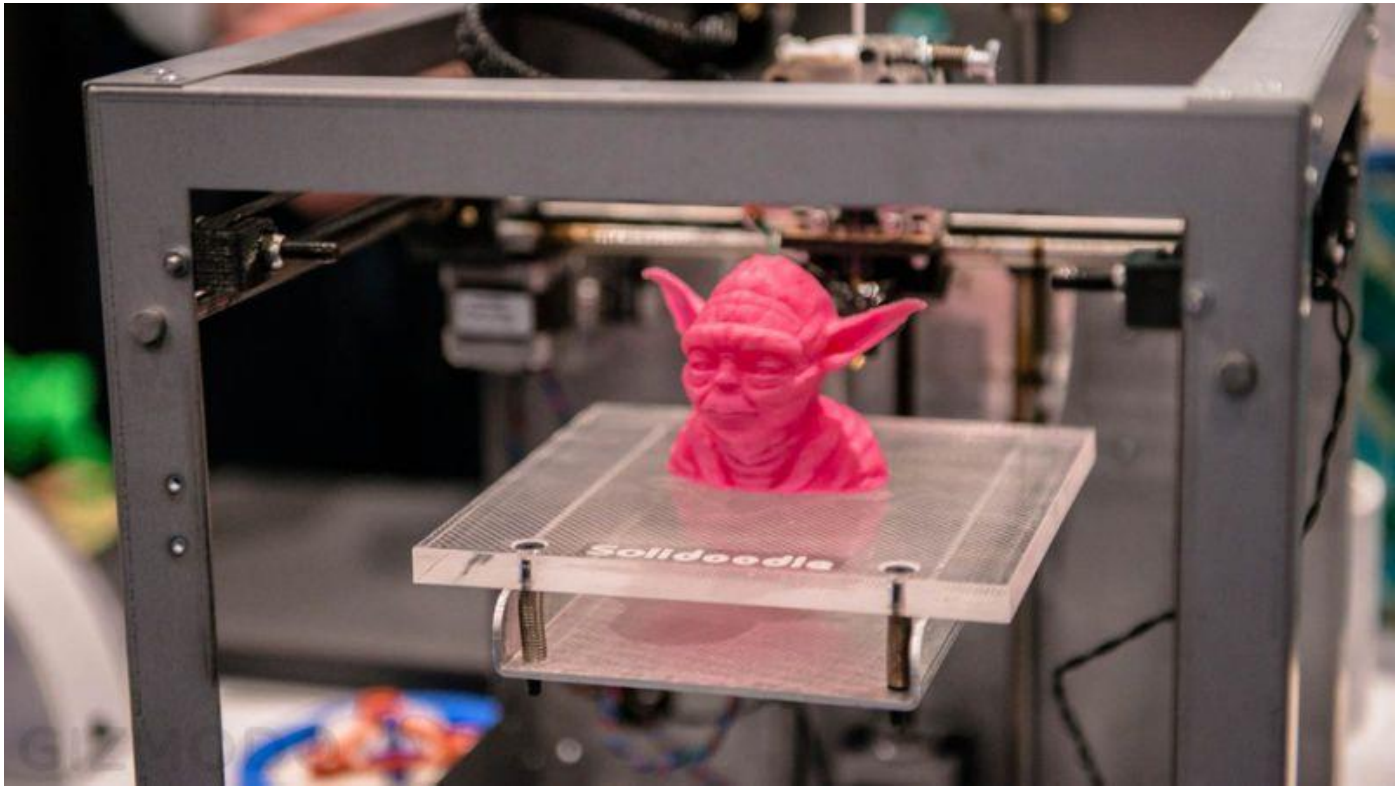
Design & Implementation – Prototyping

Paper prototype


11:15
Friday, Jan 15
Learned new words for cafe! Slide to view
> slide to unlock
CAFE
15% complete
Match each word on the left with its correct meaning on the right.
café
taza
cuchara
spoon
coffee
cup
CAFE
20% complete
Match each word on the left with its correct meaning on the right.
café
taza
cuchara
spoon
coffee
cup
✓Correct!
CAFE
LEARN NEW WORDS?
YES
NO
café

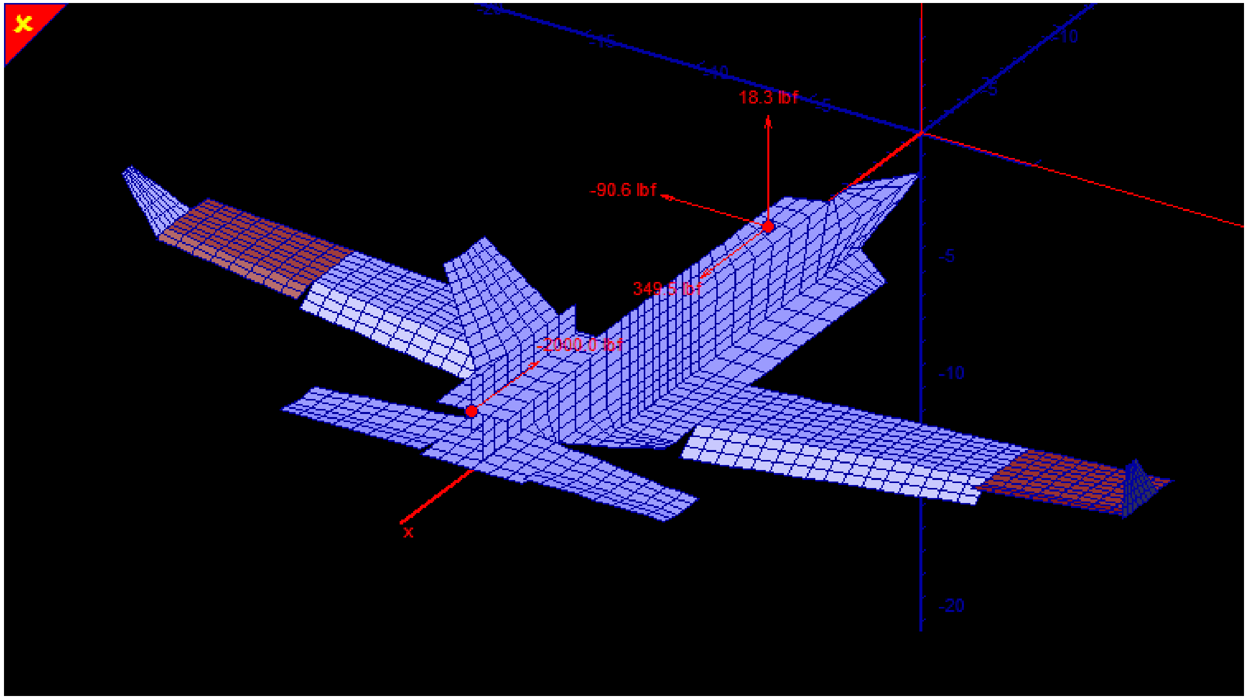
Design & Implementation – Prototyping

3D prototype



Design & Implementation – Prototyping

Digital prototype



Design & Implementation – Prototyping

Scale model (another 3D prototype)

Design & Implementation – Prototyping

Low-Fidelity prototypes are the least complete

- Quick to construct
- Easier to explore multiple versions
- Limited functionality
- Show general behaviour and interface rather than specifics
- Are a good communication aid
- Easier to criticize (this is a good thing!)
- Can overlook issues to keep testing
- Limits possible testing and comprehension

Design & Implementation – Prototyping

High-Fidelity prototypes are the most complete

- Actual working system
- Partial to complete functionality
- Easier to explore the system in-depth
- Results are closer to true system results
- Expensive
- Harder to change things at this point in the process
- Harder to criticize (a bad thing)
- Hard to keep going beyond early issues
- Management may see a high-fidelity prototype as ‘mission complete’ – “it runs, ship it!”

Design & Implementation – Learning Outcome Reflection

You should be able to

- Explain the iterative approach to usability design and implementation
- List methods available for each phase of the iterative approach
- Identify the role of usability design within the wider system development lifecycle (SDLC)

Design & Implementation – Major Assignment 1

- You have a major assessment item due (when is it due? Assessment Information on Canvas!)
- You have the skills necessary to begin your Major Assignment 1
- This assignment must be completed ***iteratively*** – because of this you cannot leave it until the due date
- In later weeks your tutor will be assessing your iterations as part of marking this assessment item

- Again: If you are not in a group in Canvas this is an issue we need to solve NOW.