COM2002/3002/6222
Human Centred Systems Design

HC Lecture 11

HCI and Software Engineering

The University Of Sheffield.

Assignment

- 100 words per question – one question max at 120 words
- Use keywords and bullet points
- My assignment is **due Monday 13th Nov 2017**

The University Of Sheffield.

Guest Lecturer: *Sebastian Conran*

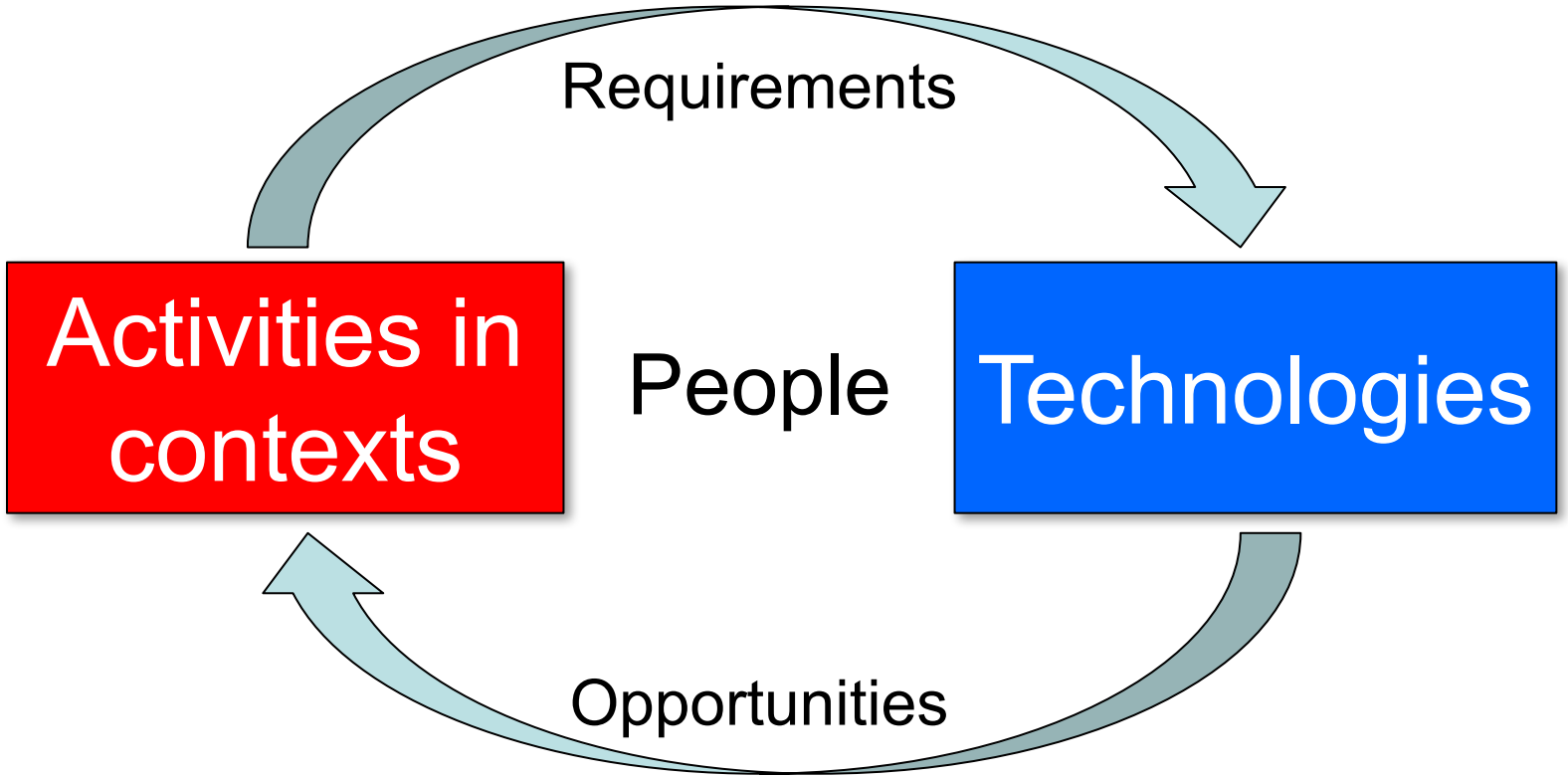
Univ. Sheffield
'*Designer in Residence*'

http://sebastianconran.com/

The University Of Sheffield.

Sebb Conran & PACT - human-centred design

"People use technologies to undertake activities in contexts"

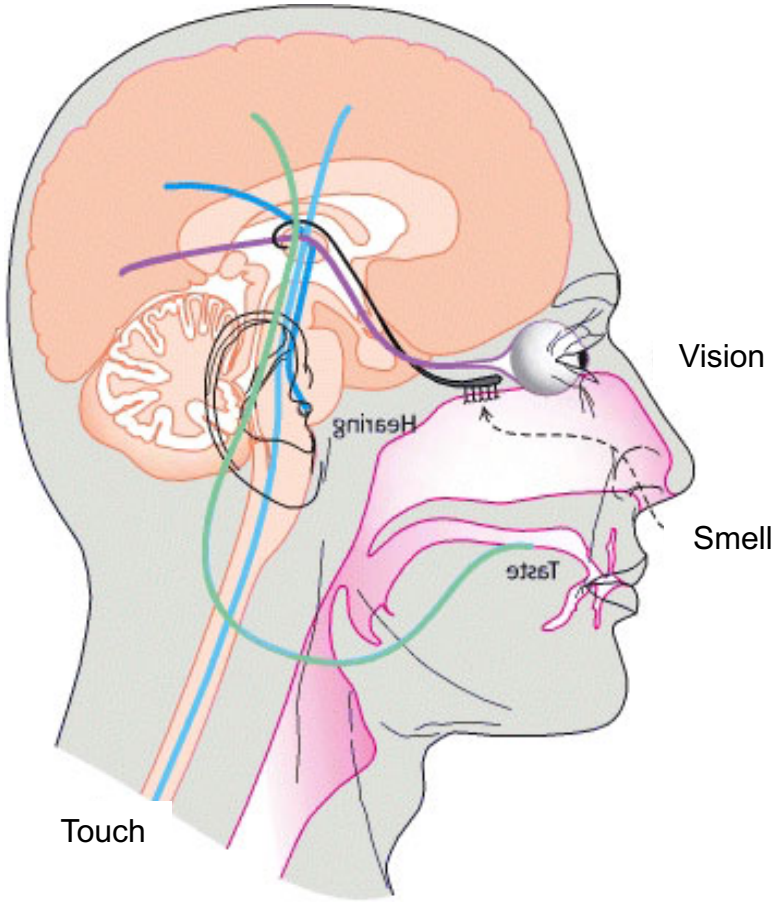


Carroll, J. M. (2002). *HCI in the New Millennium*. Harlow: Addison-Wesley.

The University Of Sheffield.

Sebb Conran & System feedback

EVALUATION



light, colours, images, video, motion, animation → Visualisation

sounds, music → Sonification Audification

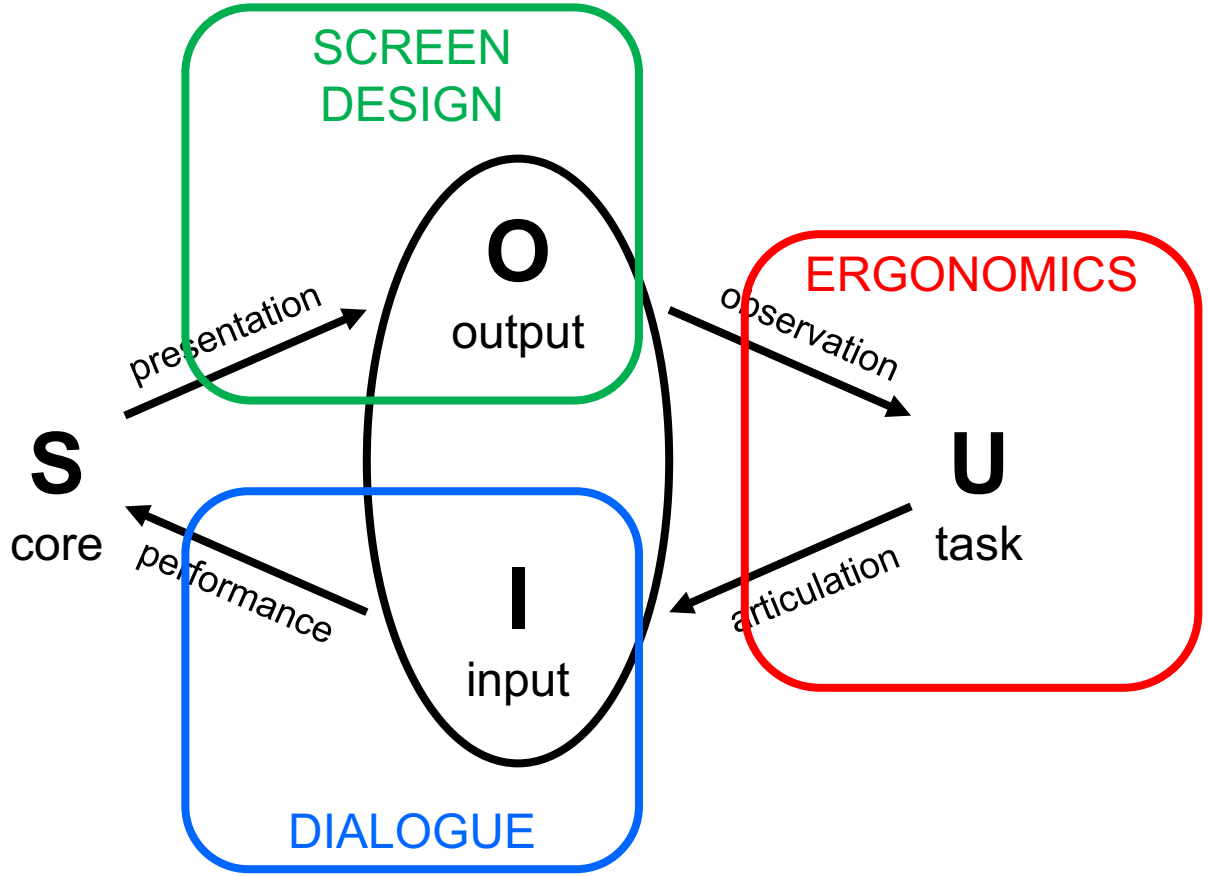
tactile, vibration, electrical, thermal → Haptic and tactile stimulus

scent, odour, pheromones → Olfactory stimulation

Interface

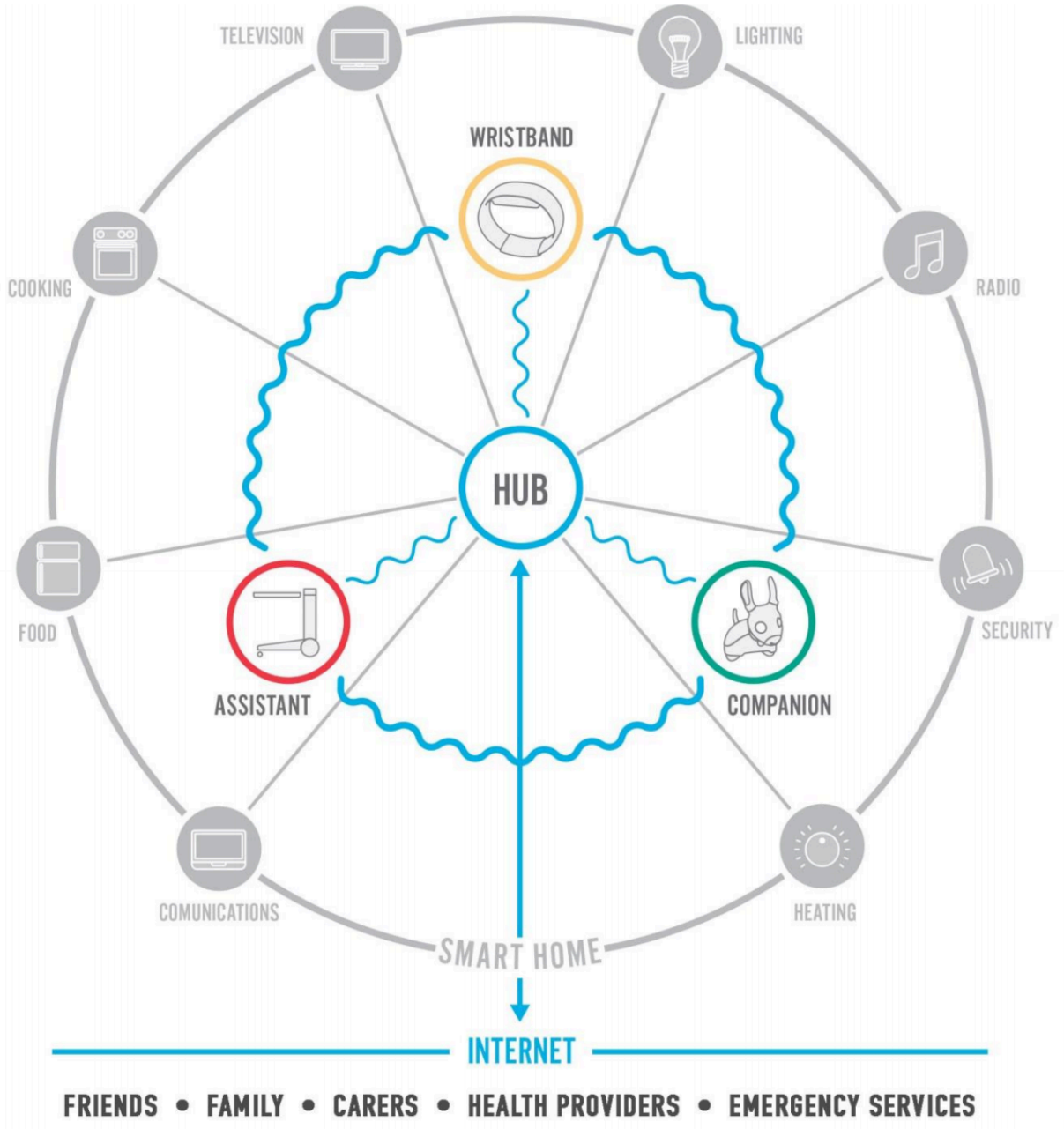
The University Of Sheffield.

Seb Conran & the 'Interaction Framework'



Abowd, G. D., & Beale, R. (1991). Users, systems and interfaces: a unifying framework for interaction. In D. Diaper & N. Hammond (Eds.), *HCI'91: People and Computers VI* (pp. 73-87). Cambridge: Cambridge University Press.

Care-Free Home System



The University Of Sheffield.


https://www.nytimes.com/2017/11/05/sports/blind-marathoner-technology.htm

This week: HCI and software engineering

Three weeks ago - - designing interaction

- Not as obvious as you think:
  - those who interact directly with the product
  - those who manage direct users
  - those who receive output from the product
  - those who make the purchasing decision
  - those who use competitor's products

- Three categories of user:
  - 'primary': *frequent hands-on*
  - 'secondary': *occasional or via someone else*
  - 'tertiary': *affected by its introduction, or will influence its purchase*
  -

The University Of Sheffield.

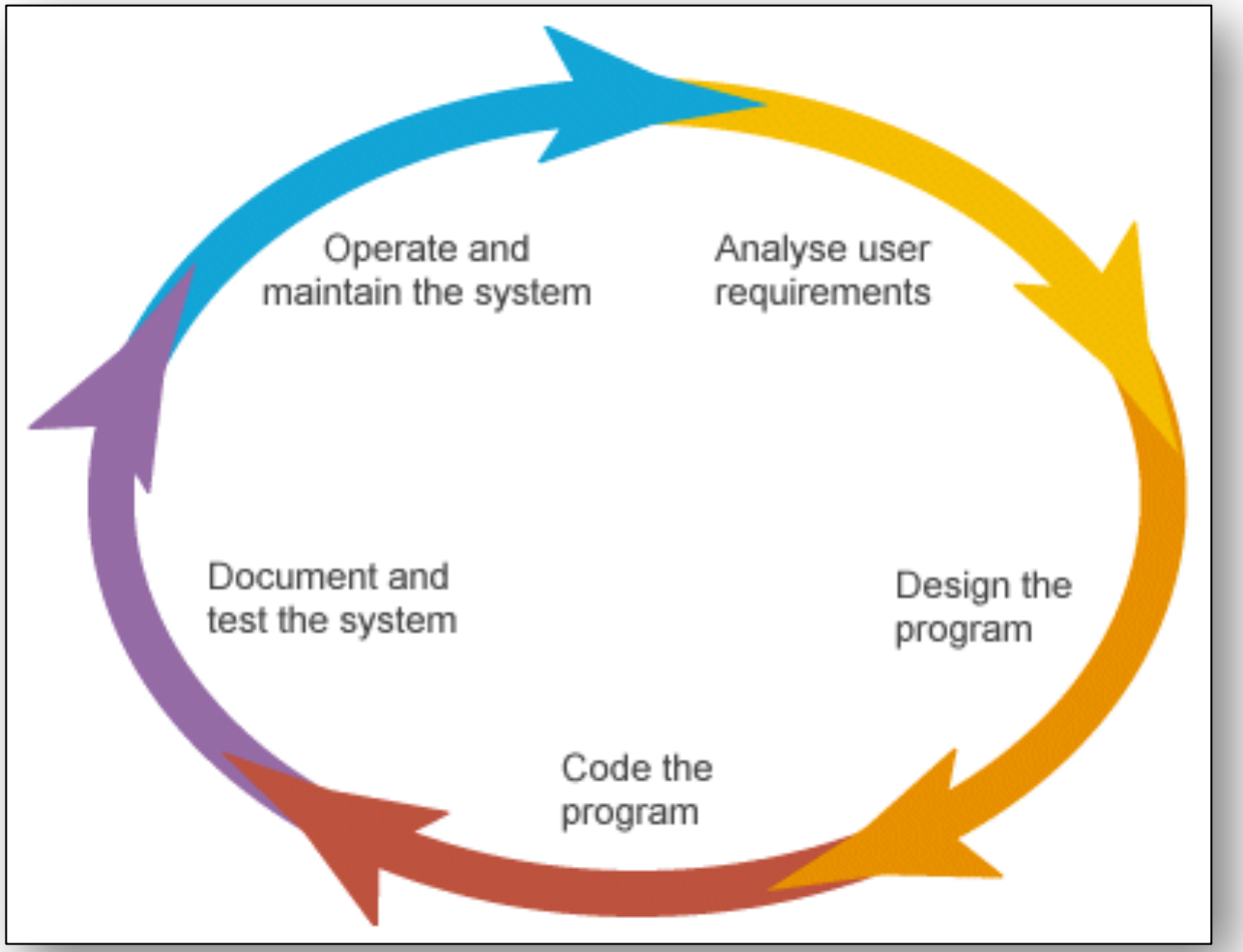
Three weeks ago - - designing interaction

- Not as obvious as you think:
  - those who interact directly with the product
  - those who manage direct users
  - those who receive output from the product
  - those who make the purchasing decision
  - those who use competitor's products
  - those who design, project manage, develop and maintain a system

- Three categories of user:
  - 'primary': *frequent hands-on*
  - 'secondary': *occasional or via someone else*
  - 'tertiary': *affected by its introduction, or will influence its purchase*
  - 'facilitating': *affected by how easy the system is to manage, and maintain*

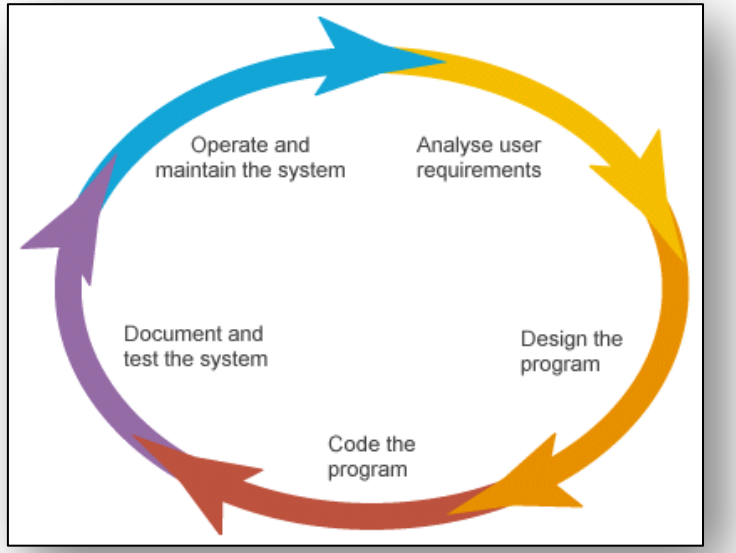
The University Of Sheffield.

The Software '*Life Cycle*'



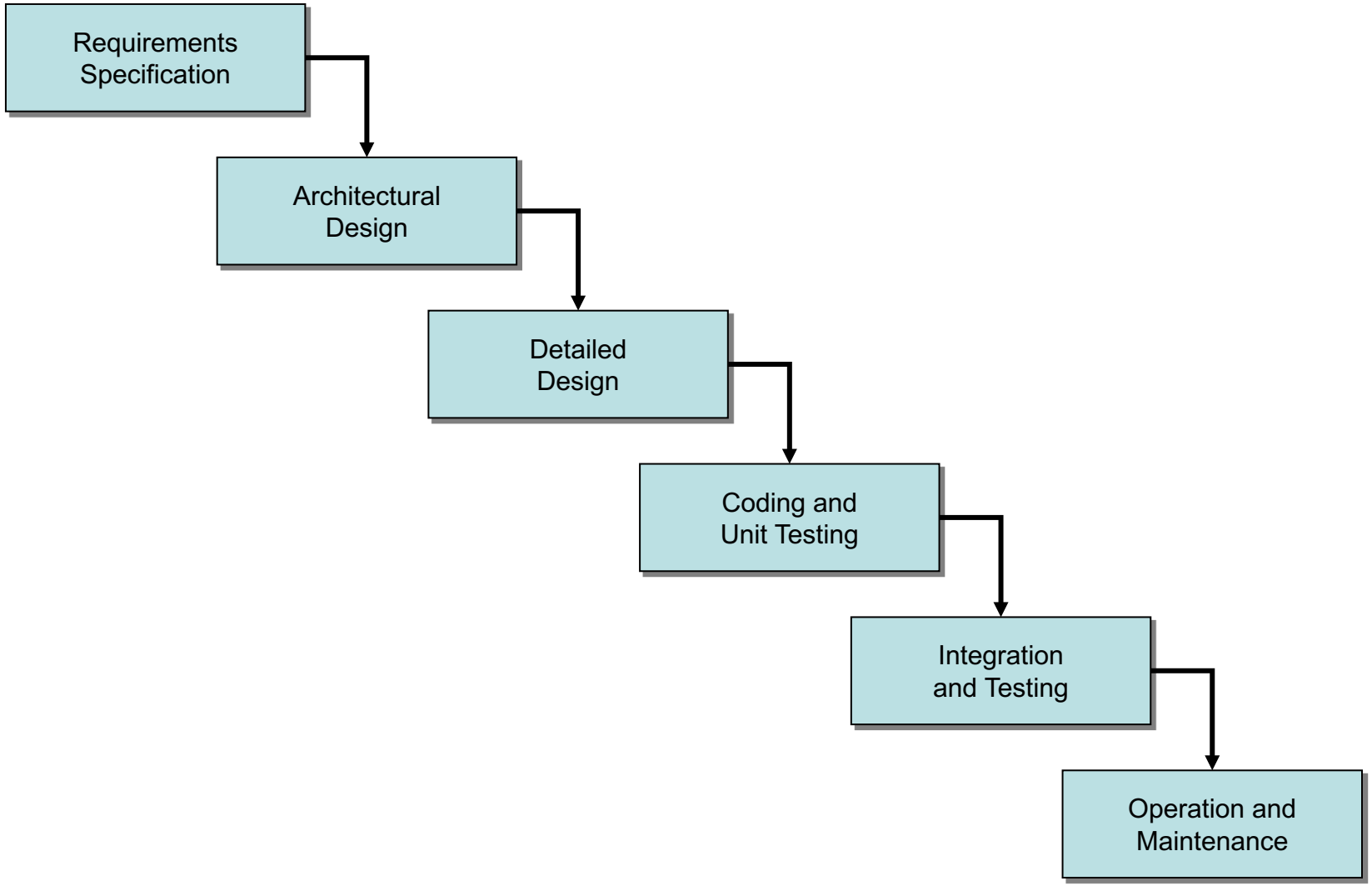
The University Of Sheffield.

The Software '*Life Cycle*'

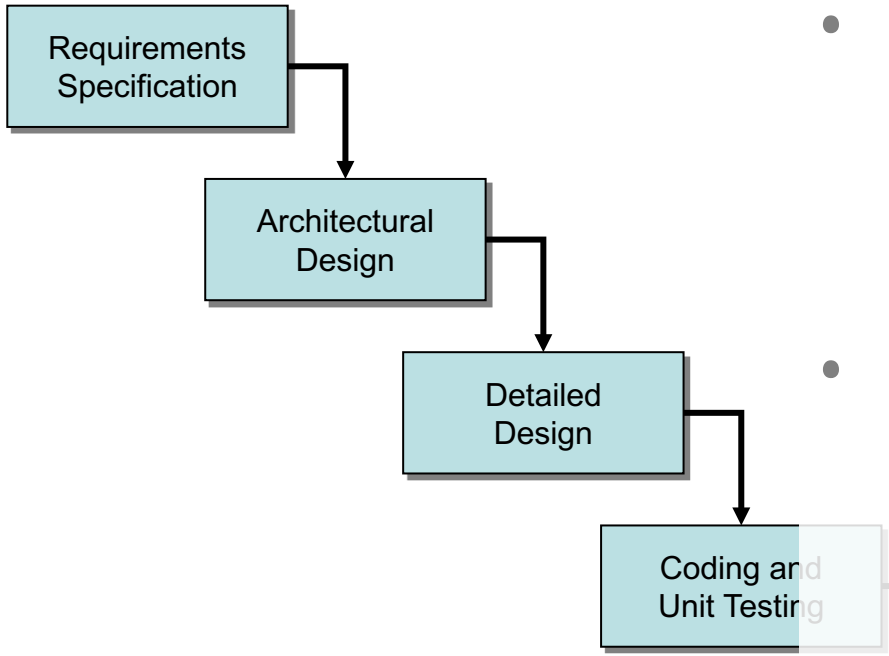


- Software engineering (*SE*) is the discipline for understanding the software design process, or 'life cycle'
- Designing for usability occurs at all stages of the life cycle, not as a single isolated activity
- Two main models for representing the software life cycle:
  - the 'waterfall model' (assumes all elements can be predefined)
  - the 'prototype model' (assumes the system must be build for the requirements to emerge)

The 'Waterfall Model'



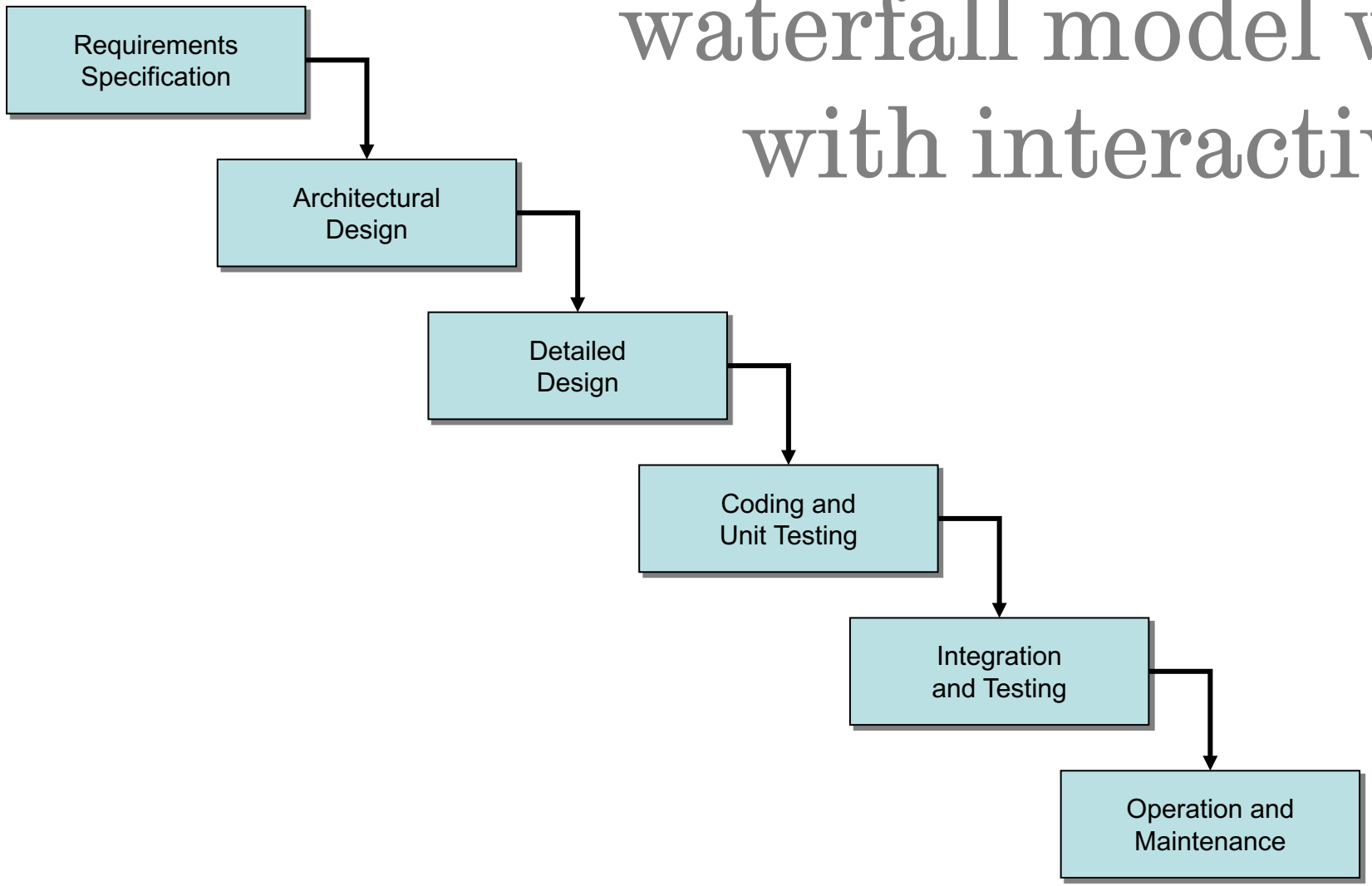
Design part of waterfall 'Life Cycle'



- Requirements specification ...
  - designer and customer try capture what the system is expected to provide
  - can be expressed in natural language or more precise languages (*such as a task analysis would provide*)
- Architectural design ...
  - high-level description of how the system will provide the services required
  - factor system into major components and how they are interrelated
  - needs to satisfy both functional and non-functional requirements
- Detailed design ...
  - refinement of architectural components and interrelations to identify modules to be implemented separately
  - governed by the non-functional requirements

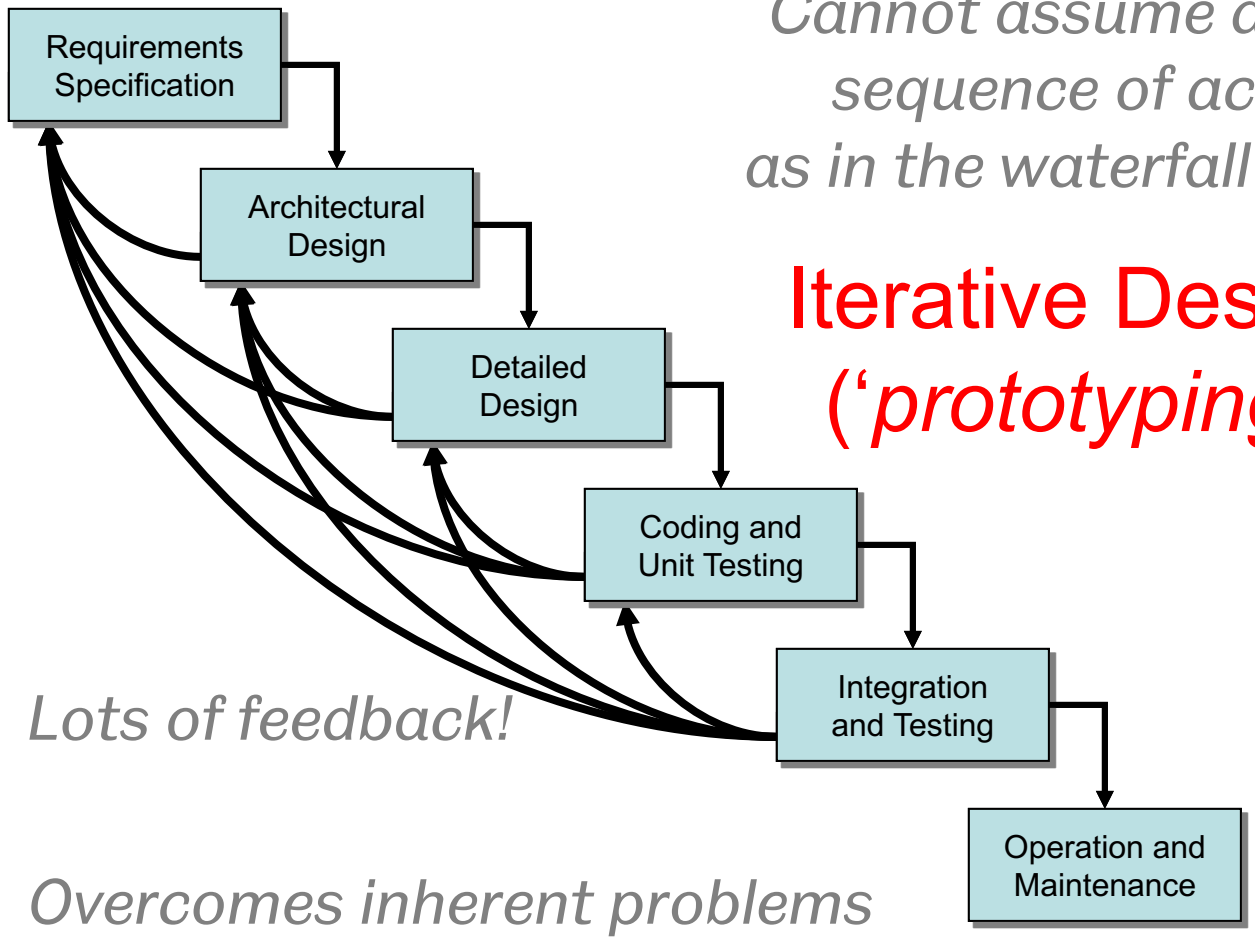
The University Of Sheffield.

How should we modify the waterfall model when dealing with interactive design?



The University Of Sheffield.

The 'Life Cycle' for Interactive Systems



Cannot assume a linear sequence of activities as in the waterfall model

Iterative Design ('*prototyping*')

Lots of feedback!

Overcomes inherent problems of incomplete requirements

The University Of Sheffield.

What to Prototype?

- Technical issues
- Work flow
- Task design
- Screen layouts
- Information displays
- Difficult / controversial / critical issues

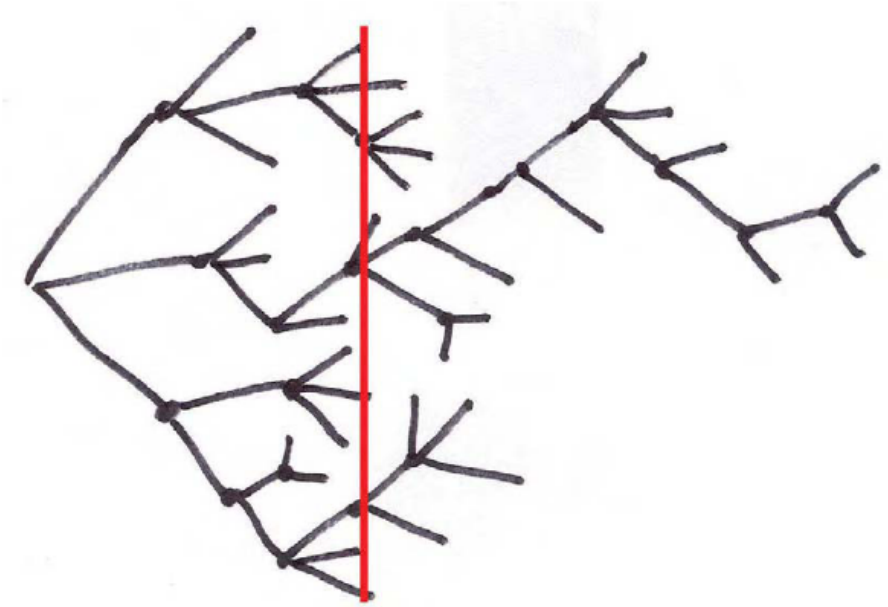
Design
"Getting the right design"

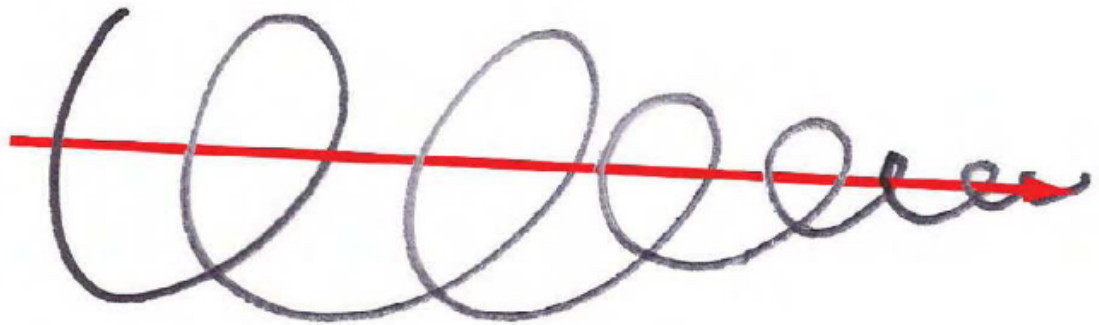
v

Prototyping
"Getting the design right"

Bill Buxton

Prototyping





Design
"Branching Exploration"

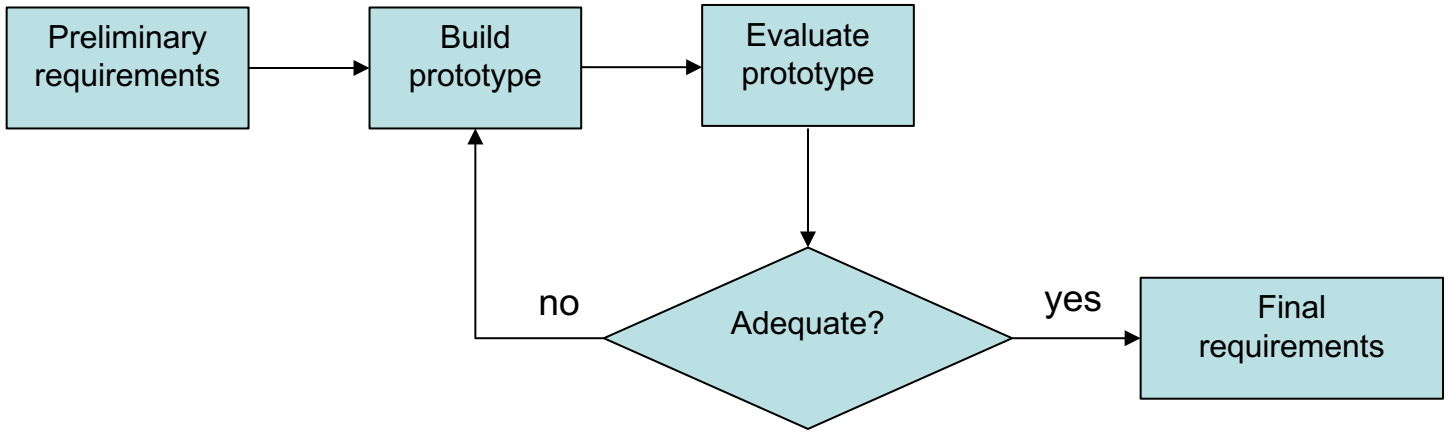
v

Prototyping
"Incremental iterative refinement"

Bill Buxton

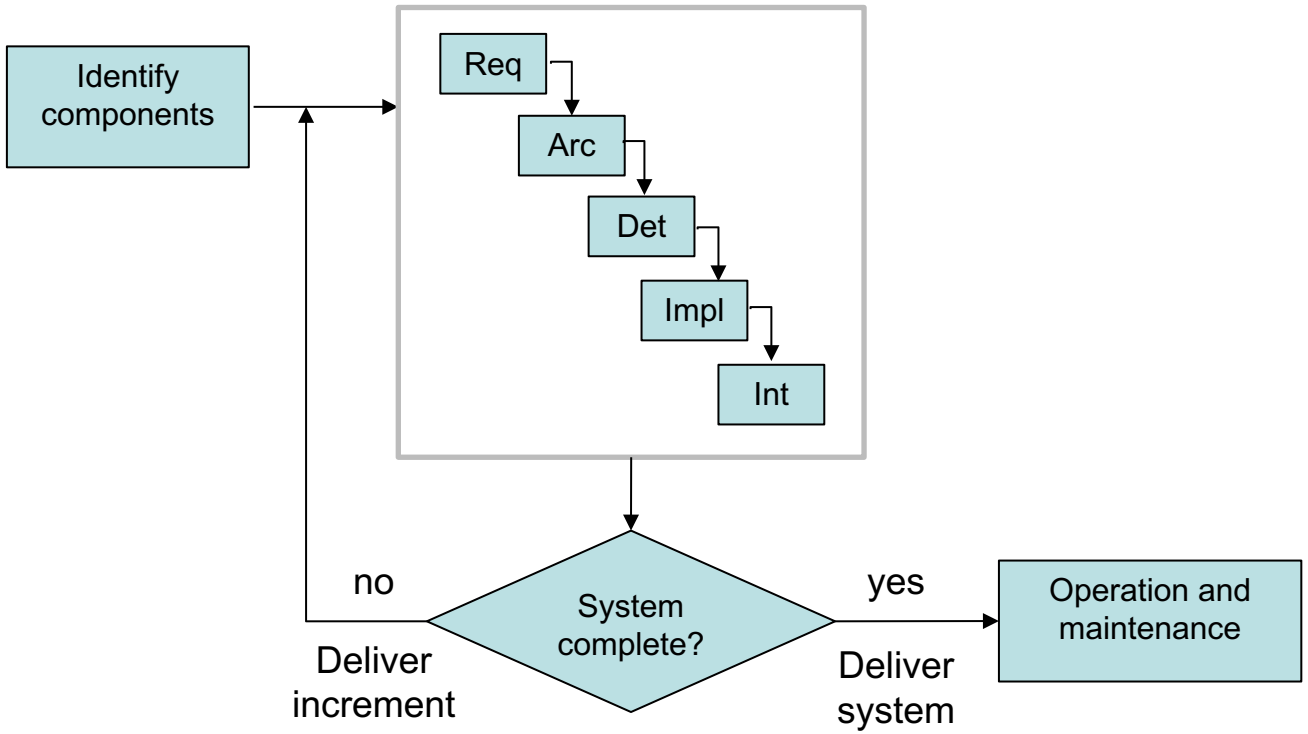
1) 'Throw-away' (*rapid*) Prototyping

- sequence of *independent* prototypes
- each prototype is evaluated against requirements
- each prototype is discarded and a new one built
- iterate towards final requirements



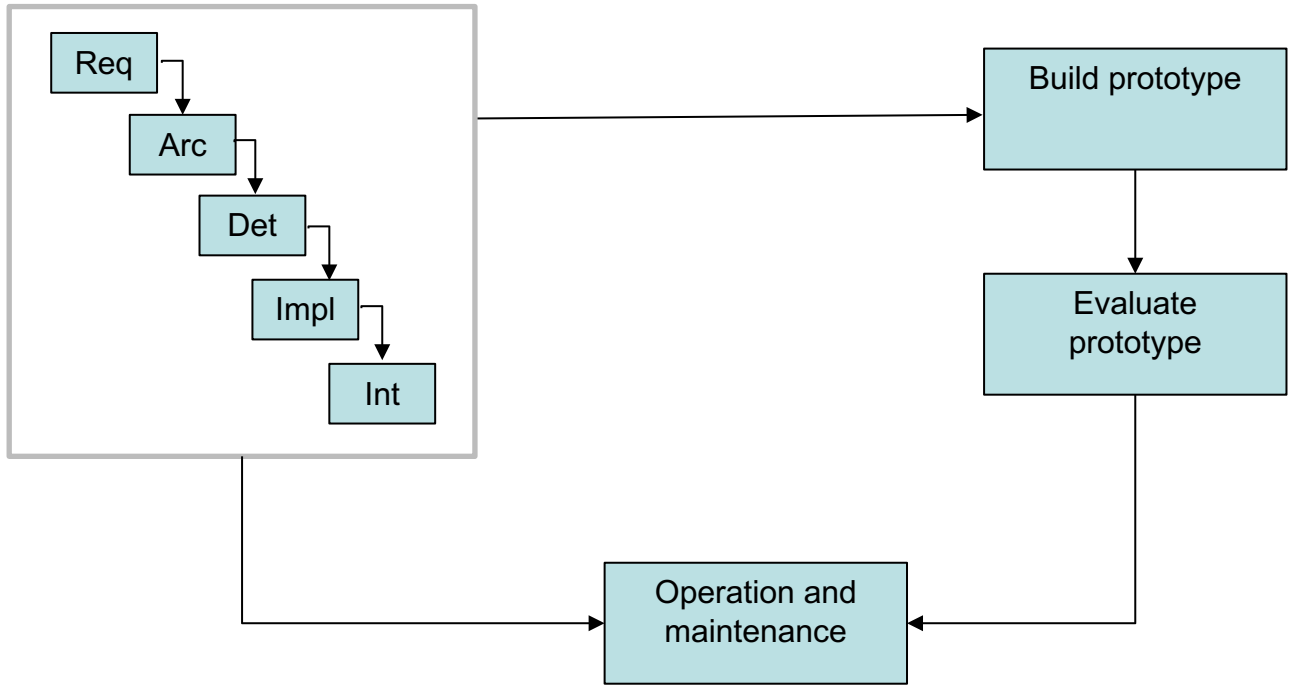
2) 'Incremental' Prototyping

- final product built from separate components
- multiple product releases
- each release includes one more component



3) 'Evolutionary' Prototyping

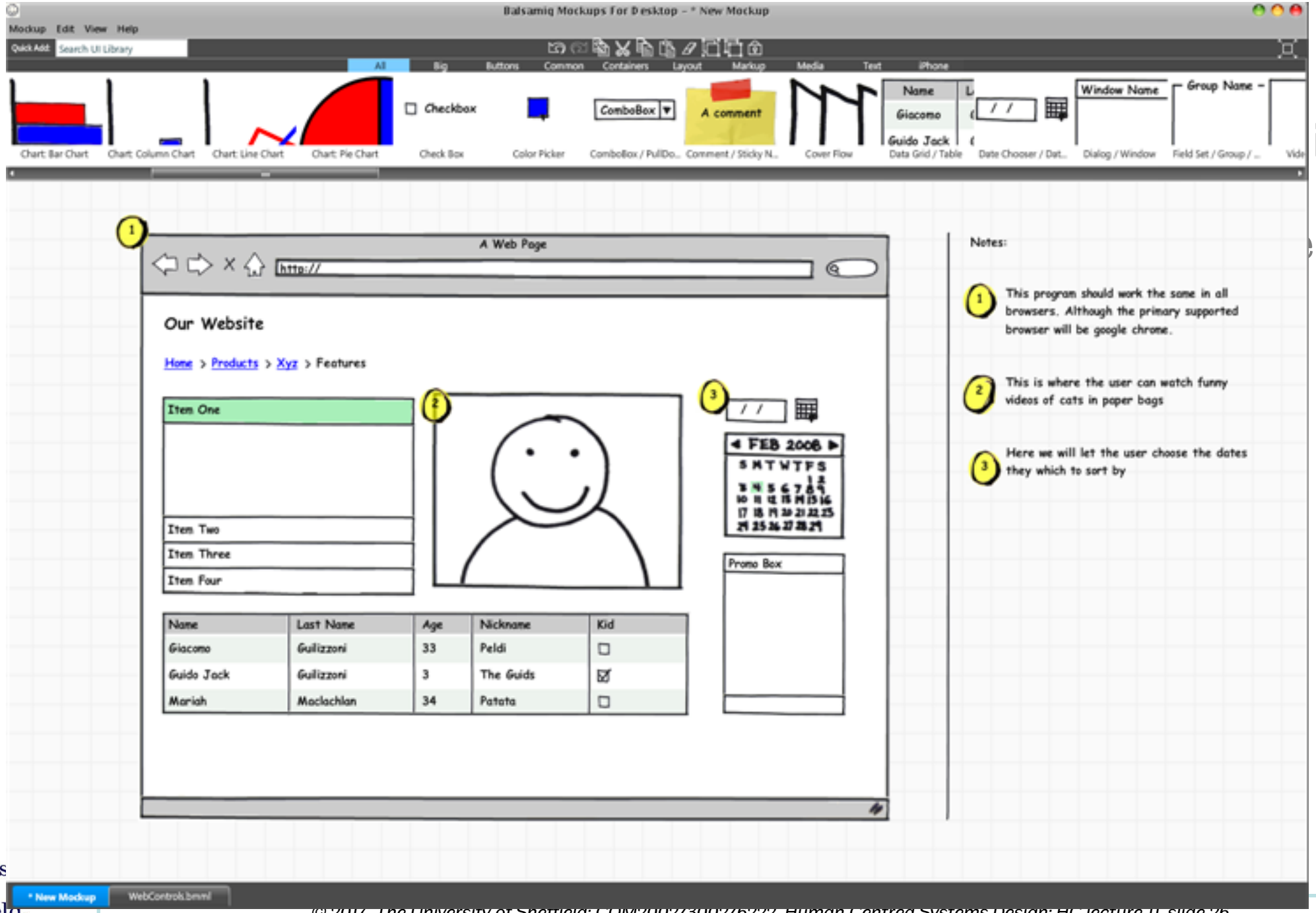
– prototypes not discarded, but form basis of the next iteration of the design



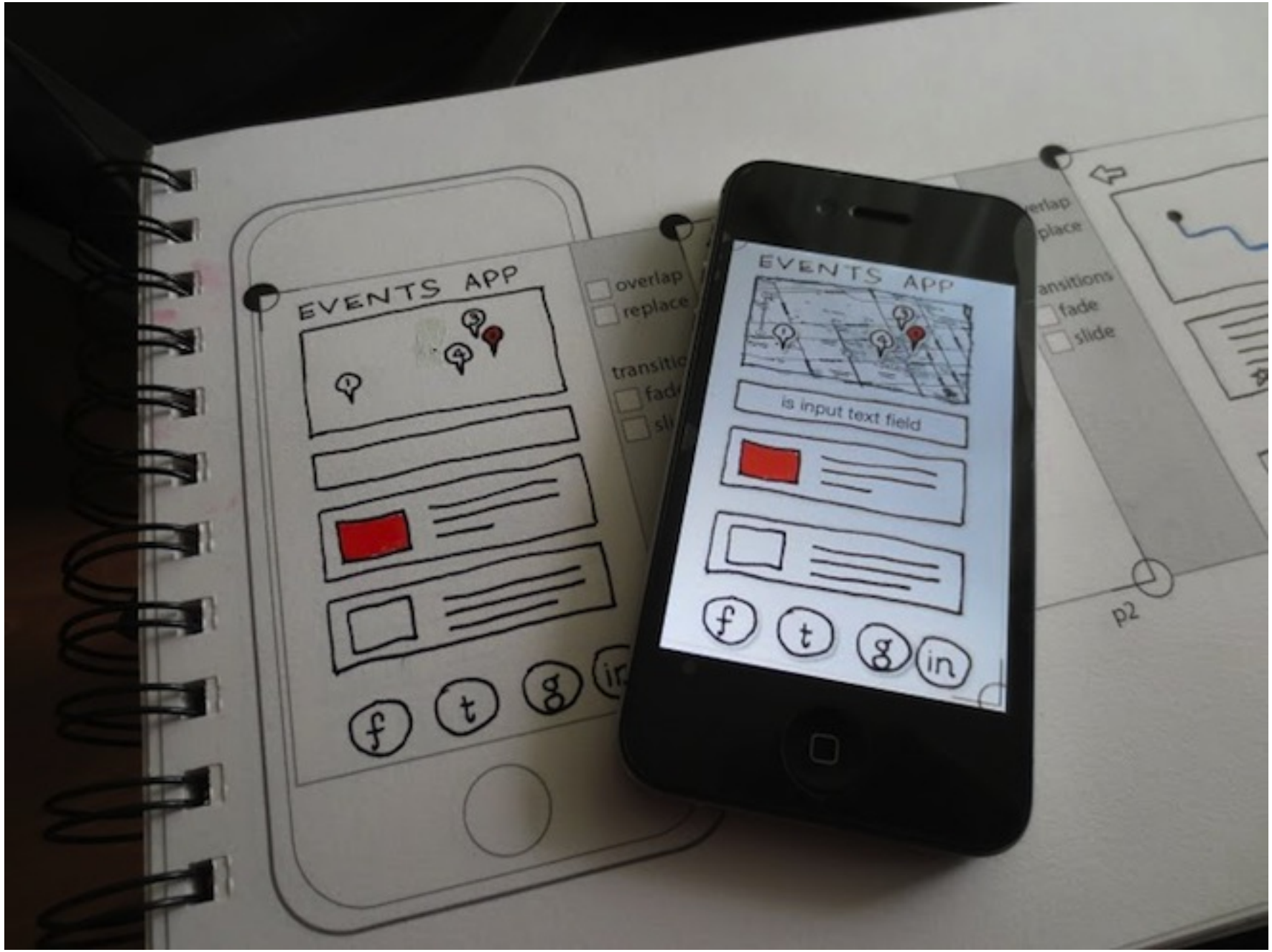
The University Of Sheffield.

What form can a prototype take?

- A series of screen sketches
- A 'storyboard' (*i.e. a cartoon-like series of scenes*)
- A PowerPoint slide show
- A video simulating the use of a system
- A lump of wood
- A cardboard mock-up
- A piece of software with limited functionality written in the target language (*or in another language*)


Balsamiq Mockups For Desktop - * New Mockup
Mockup Edit View Help
Quick Add: Search UI Library
All Big Buttons Common Containers Layout Markup Media Text iPhone
Chart: Bar Chart
Chart: Column Chart
Chart: Line Chart
Chart: Pie Chart
Checkbox
Check Box
Color Picker
ComboBox
ComboBox / PullDo...
A comment
Comment / Sticky N...
Cover Flow
Name
Giacomo
Guido Jack
Data Grid / Table
Date Chooser / Dat...
Window Name
Dialog / Window
Group Name
Field Set / Group / ...
1
A Web Page
http://
Our Website
Home > Products > Xyz > Features
Iten One
Iten Two
Iten Three
Iten Four
2
3
FEB 2008
Promo Box
Name
Last Name
Age
Nickname
Kid
Giacomo
Guilizzoni
33
Peldi
Guido Jack
Guilizzoni
3
The Guids
Mariah
Maclachlan
34
Patata
Notes:
1 This program should work the same in all browsers. Although the primary supported browser will be google chrome.
2 This is where the user can watch funny videos of cats in paper bags
3 Here we will let the user choose the dates they which to sort by
* New Mockup
WebControls.bmml

imate
of
em



The University Of Sheffield.


VISY
PAPER

Available on:

Computer | Tablet | Smartphone | Smartwatch | Television | At home | In the car | On the move

Human

Choice of gender/personalised

Professional, but non-clinical

Expressive (body language)

Set in a relaxing looking environment

Option of voice only

Regional accent, like a 'Look North' news reader

Well projected, with no slang

Changes tone & sounds human

Sounds friendly, caring, kind & calm

Option for no voice, and text only.

Features:

Panic mode (if having exacerbation)

Reminders e.g. to eat/drink

For relaxation/sleep

Link to advice/ support groups

Family/carer involvement

Contact for housebound people with COPD

Linked to GP/Therapist/Pharmacy

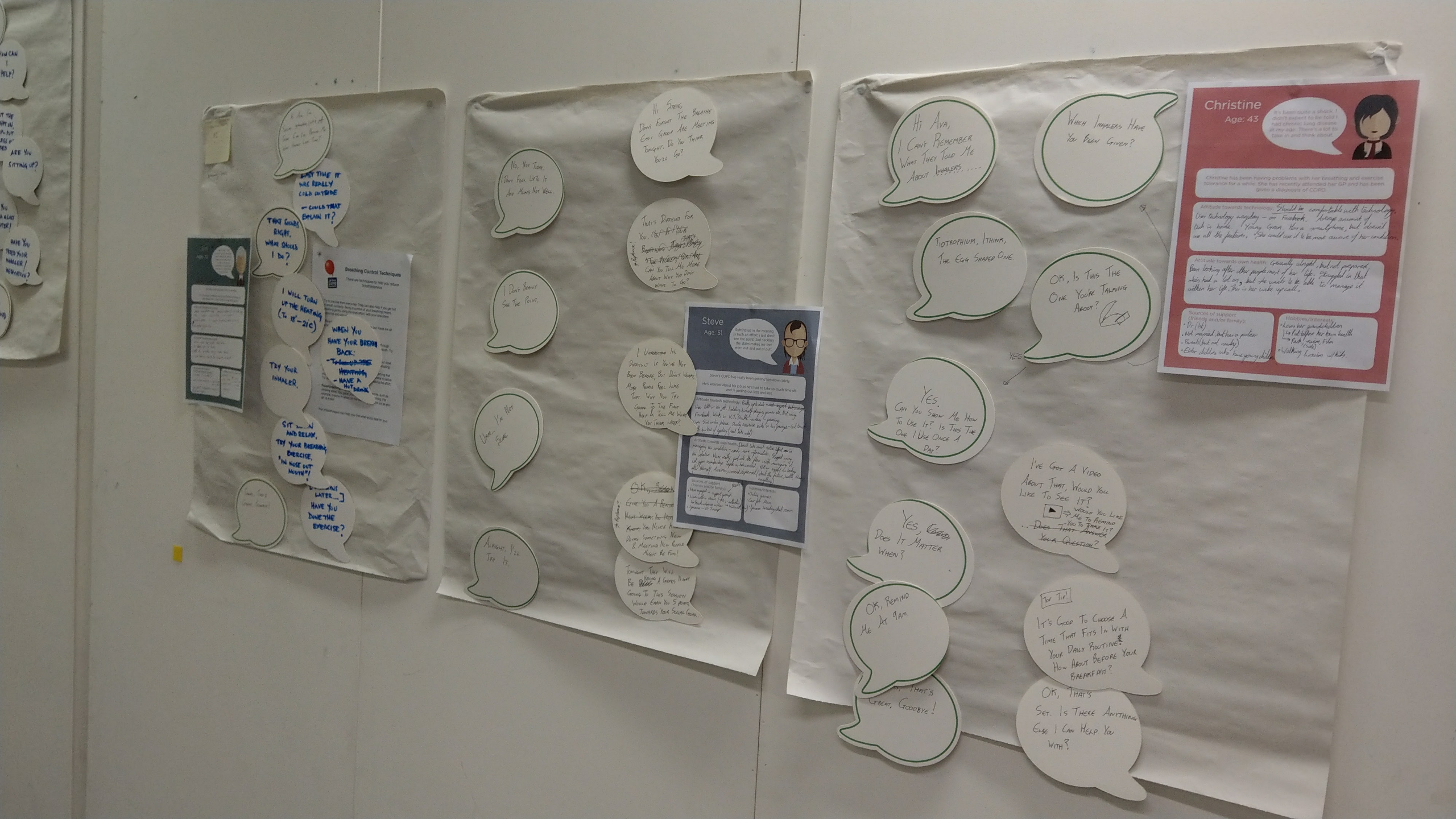
Care/Appointment Planner

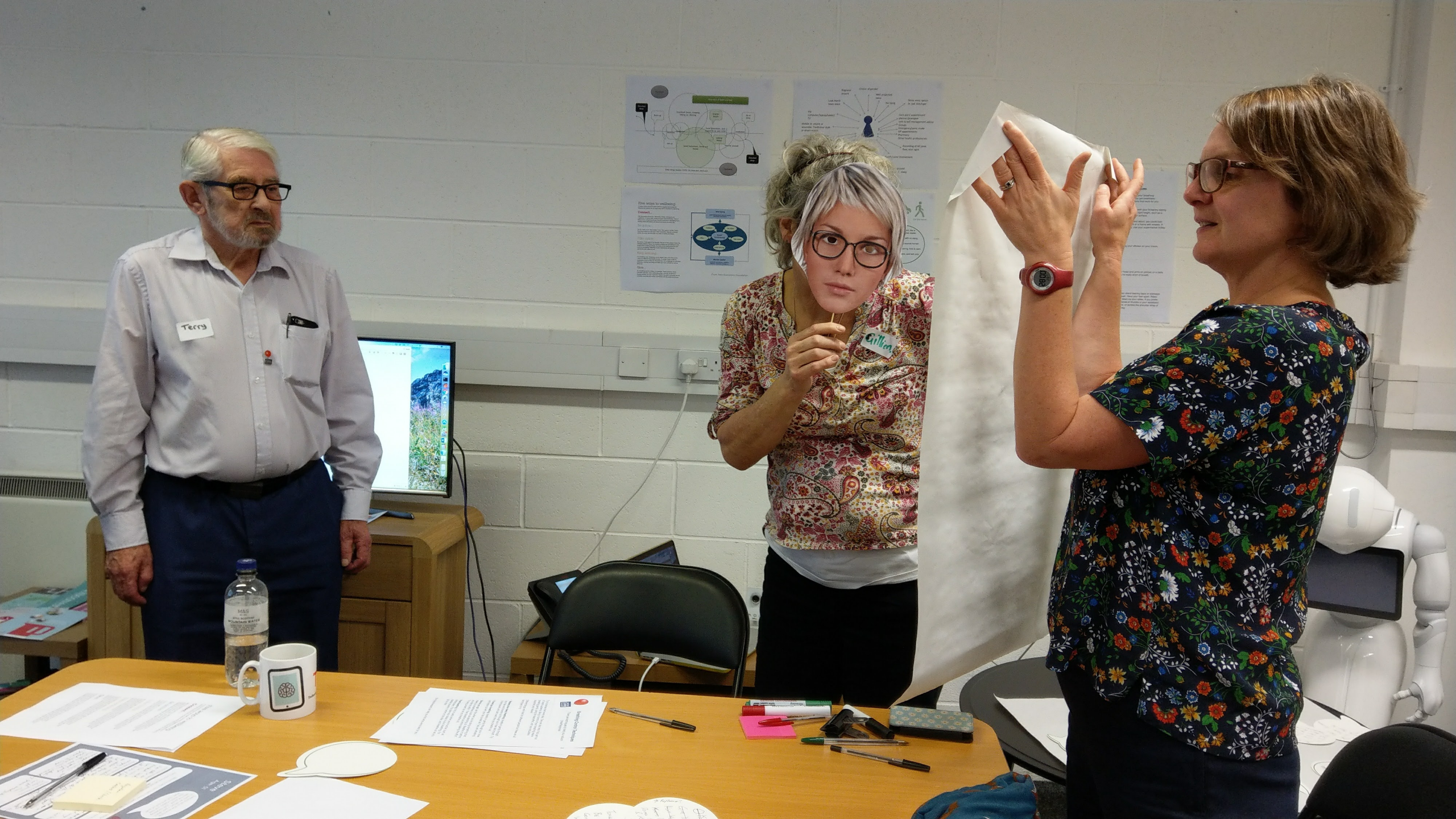
Access results/scans

Tool to help you think & interact with GP/Nurse (i.e. remember questions, link to health record)

Recording of peak flow, BP, tracks exercise, etc

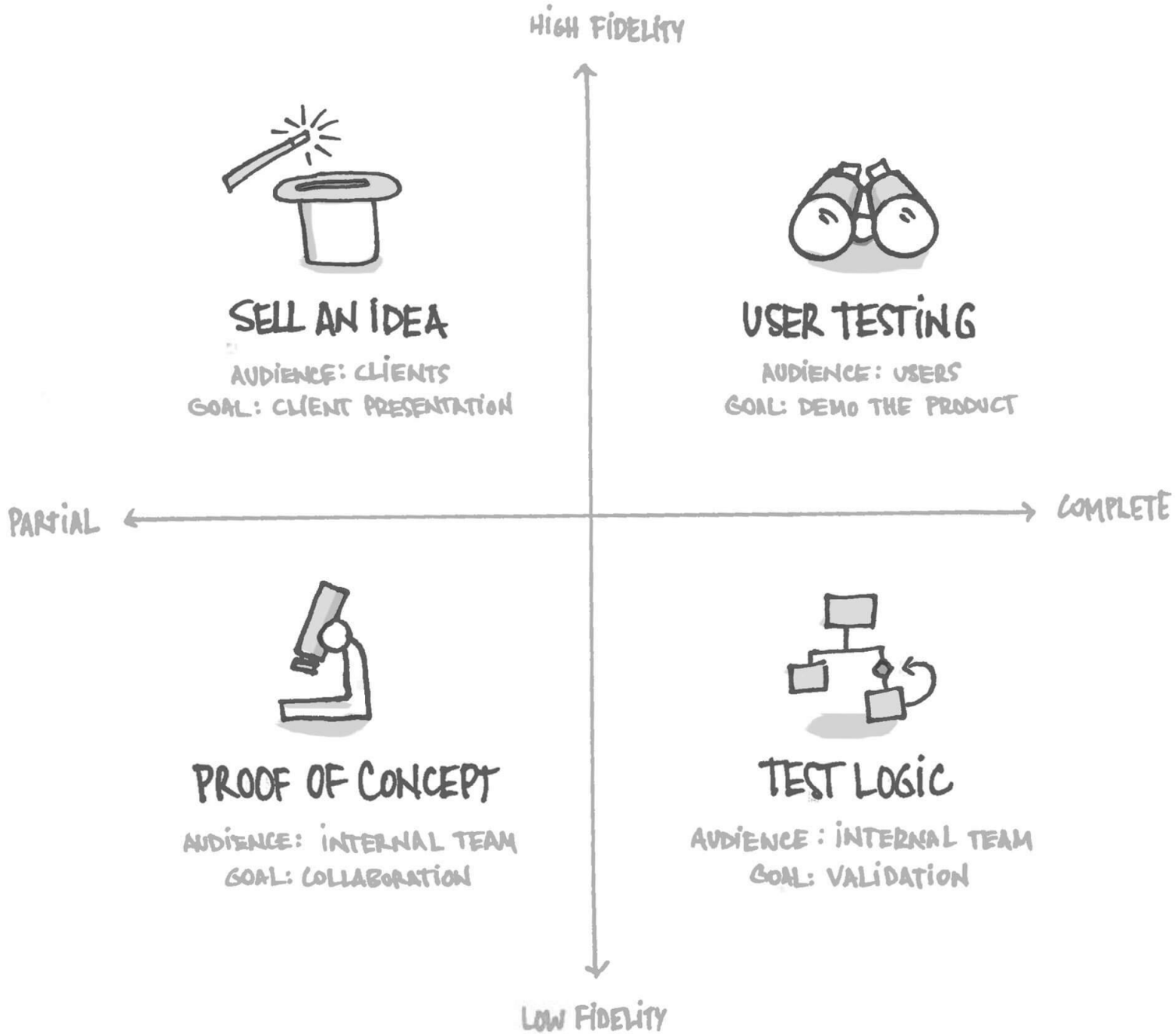

Gillian
Emily
David
Christine
Age 43
Five Ways to Wellbeing
Connect
Be Active
THAN


Breathing Control Techniques
Steve
Age: 51
Christine
Age: 43
Hi Ava, I Can't Remember What They Told Me About Inhalers, ...
Which Inhalers Have You Been Given?
Tiotrophium, Ithinh, The Egg Shaped One.
Ok, Is This The One You're Talking About?
Yes. Can You Show Me How To Use It? Is This The One I Use Once A Day?
I've Got A Video About That, Would You Like To See It?
Yes, Does It Matter When?
Ok, Remind Me At 9am
Ok, That's Great, Goodbye!
For Tip!
It's Good To Choose A Time That Fits In With Your Daily Routine. How About Before Your Breakfast?
Ok, That's Set. Is There Anything Else I Can Help You With?
Hi Steve, Don't Forget The Breathe Easy Group Are Meeting Tonight. Do You Think You'll Go?
No, Not Today. I Don't Feel Up To It And Always Not Well.
That's Difficult For You. Can You Tell Me More About Why You Don't Want To Go?
I Don't Really See The Point.
I Understand It's Difficult If You've Not Been Before, But Don't Worry, Most People Feel Like That. Why Not Try Coming To The First One & Tell Me What You Think After?
Alright, I'll Try It.
Tonight There Will Be Games & Games Night Going To This Session Would Earn You 5 Points Towards Your Social Goal.
Any Time It Was Really Cold Outside - Could That Explain It?
That Goes Right, When Should I Do?
I Will Turn Up The Heating (To 18°-21C)
Try Your Inhaler.
Sit ... And Relax, Try Your Breathing Exercise. "In Nose Out Mouth"!
... Later...] Have You Done The Exercise?
Are You Sitting Up?
Have You Tried Your Inhaler / Medication?


Terry
Gillian

Prototyping Techniques

- 'Low-fidelity' (*Lo-Fi*) prototyping
  - often termed 'paper prototypes' (*since that is what they are usually made from*)
  - focused on the broad underlying design ideas
  - designed to be produced quickly (*and thrown away as quickly*)
  - capture very early design thinking
  - most usual form is a series of screenshots
- 'High-fidelity' (*Hi-Fi*) prototyping
  - similar in look and feel (*but perhaps not functionality*) to the anticipated final product
  - useful for detailed evaluation of the main design elements
  - generally developed well into the project
  - often constitutes a crucial stage in client acceptance



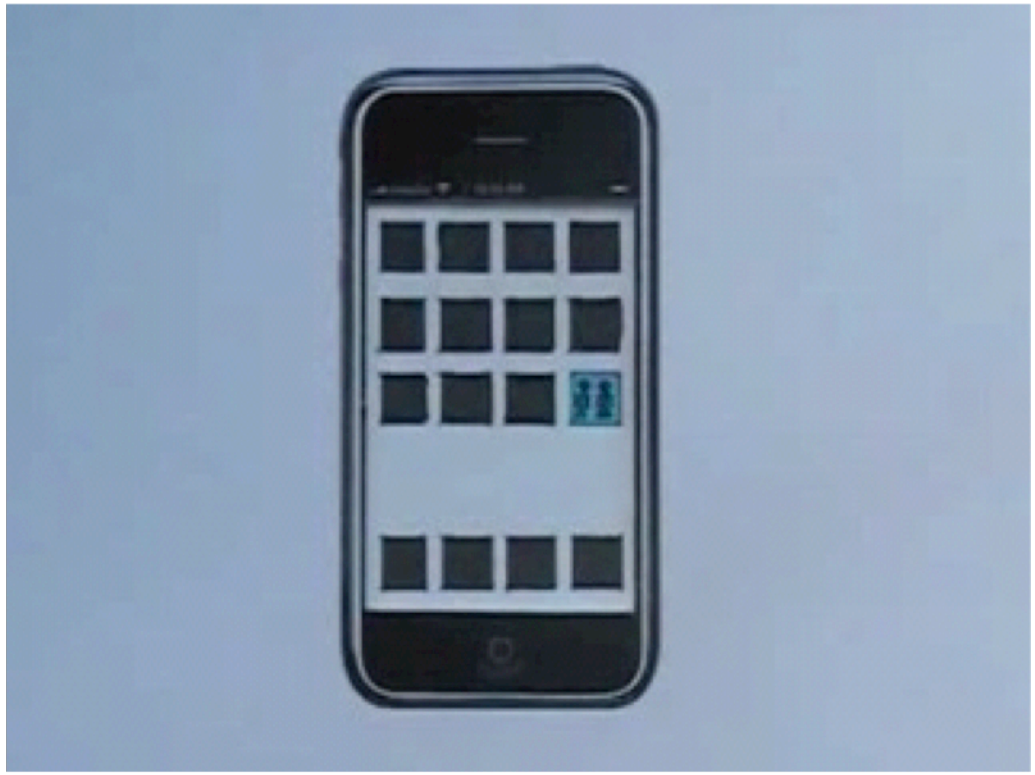
https://www.smashingmagazine.com/2016/06/picking-the-best-prototyping-software-for-your-project/

‘Lo-Fi’ Prototyping

- Screen sketches
  - permanent features are drawn on card
  - dynamic items (*such as dialogue boxes or menus*) use the cards or Post-It notes
  - overlays of acetates can simulate dynamic features (*or allow people to write comments using wipe-off pens*)
- ‘Storyboards’
  - a simple cartoon-like structure
  - need not be computer-based
  - can be animated
- Limited functionality simulations
  - some part of system functionality provided by designers
  - ‘Wizard of Oz’ technique

The University Of Sheffield.

‘Lo-Fi’ Prototyping



Paper prototyping is used to check the use process of the iPhone app

http://www.youtube.com/watch?v=6TbyXq3XHSc

‘Storyboarding’

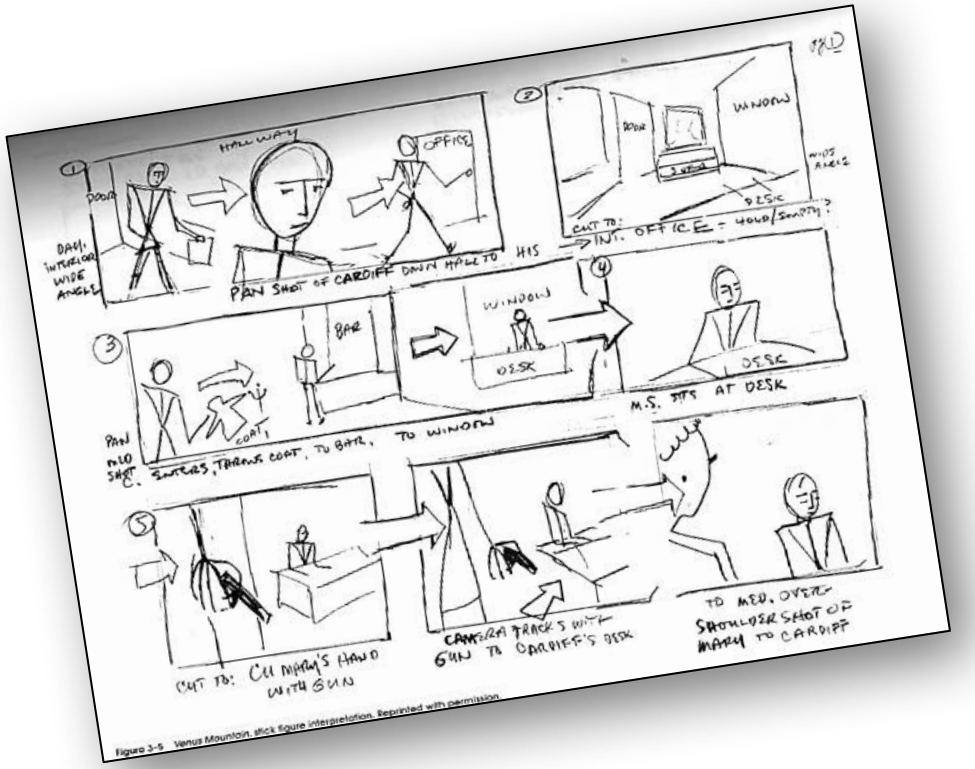
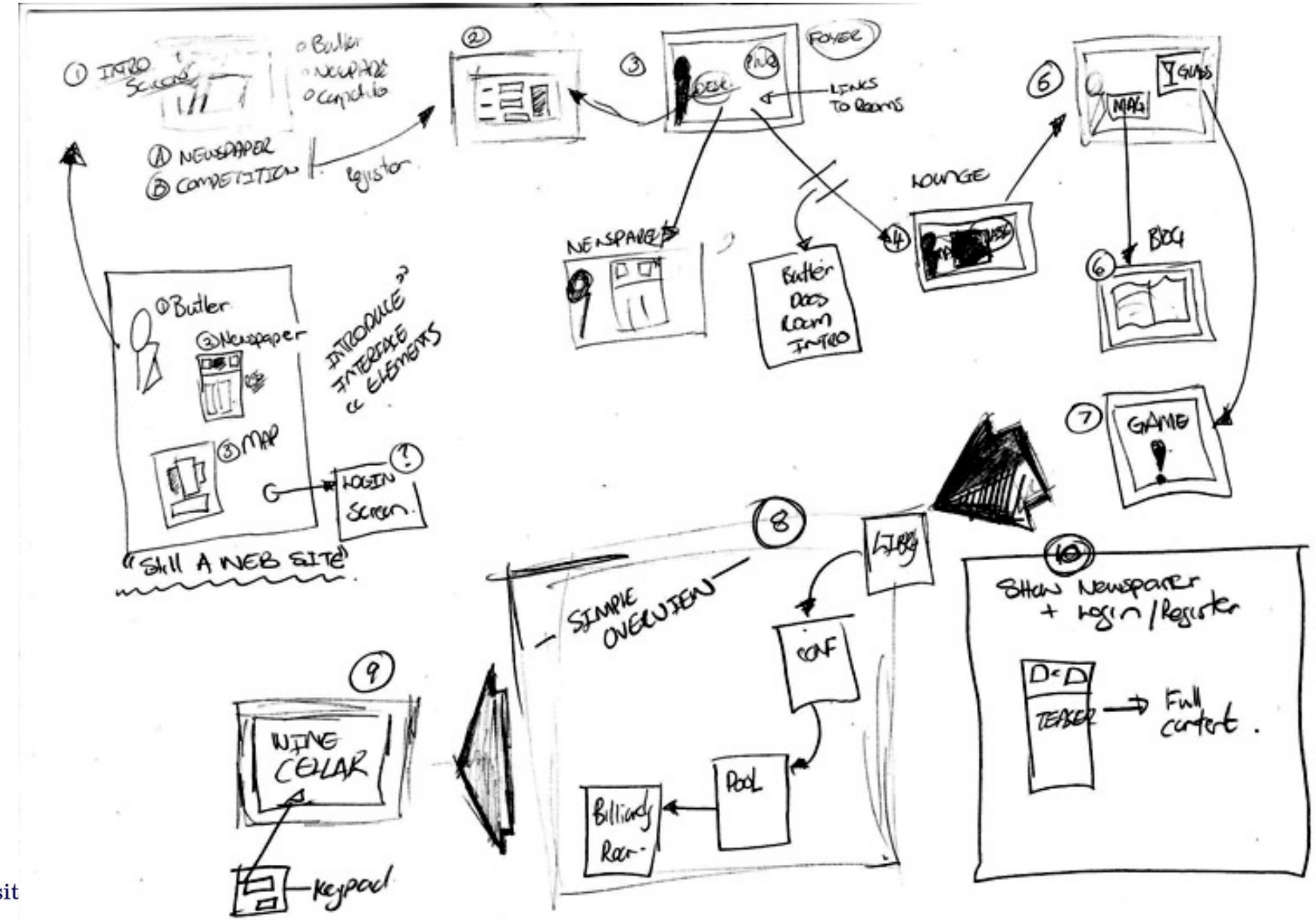
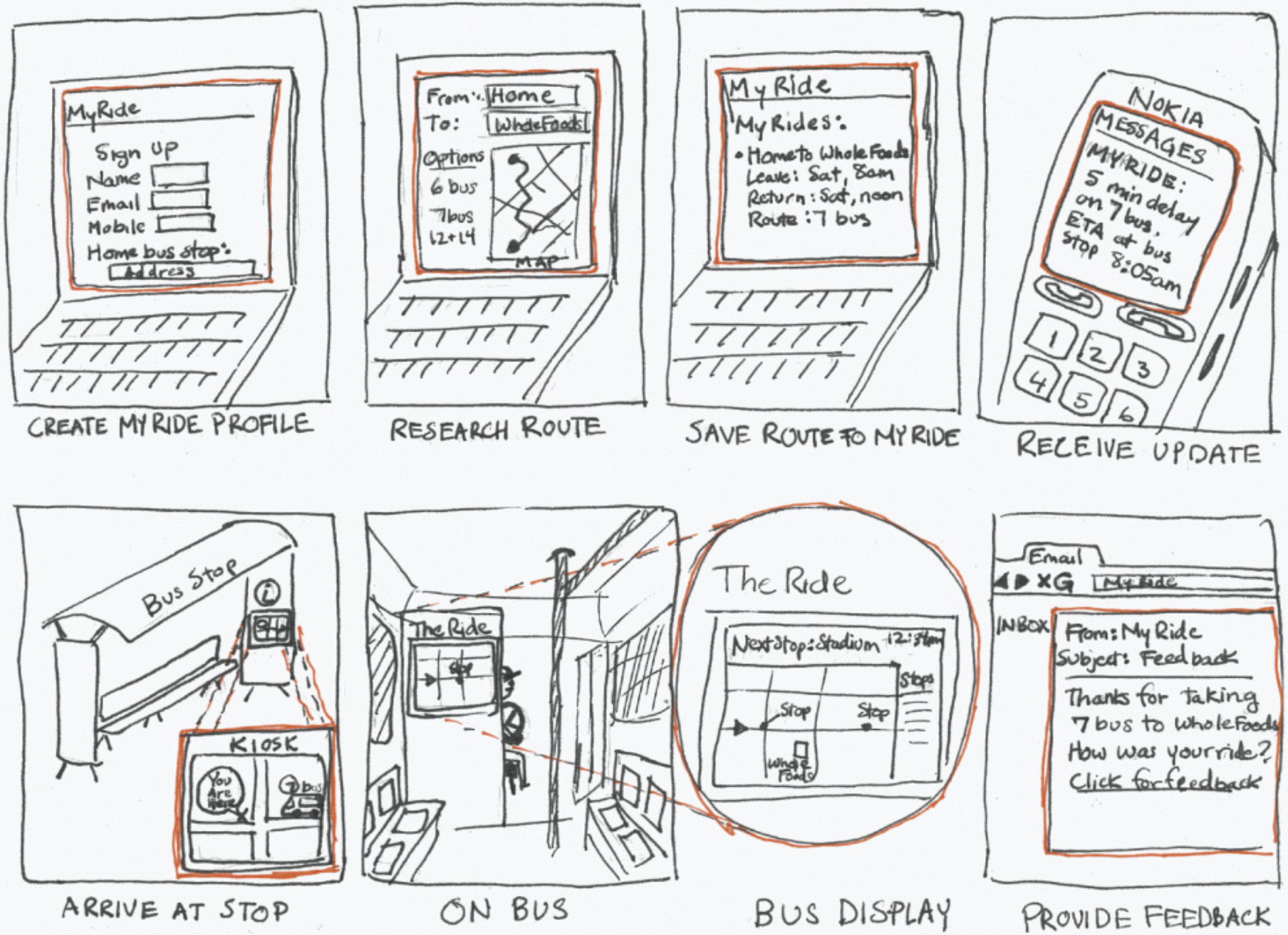


Figure 3-5 Venus Mountain stick figure interpretation. Reprinted with permission.

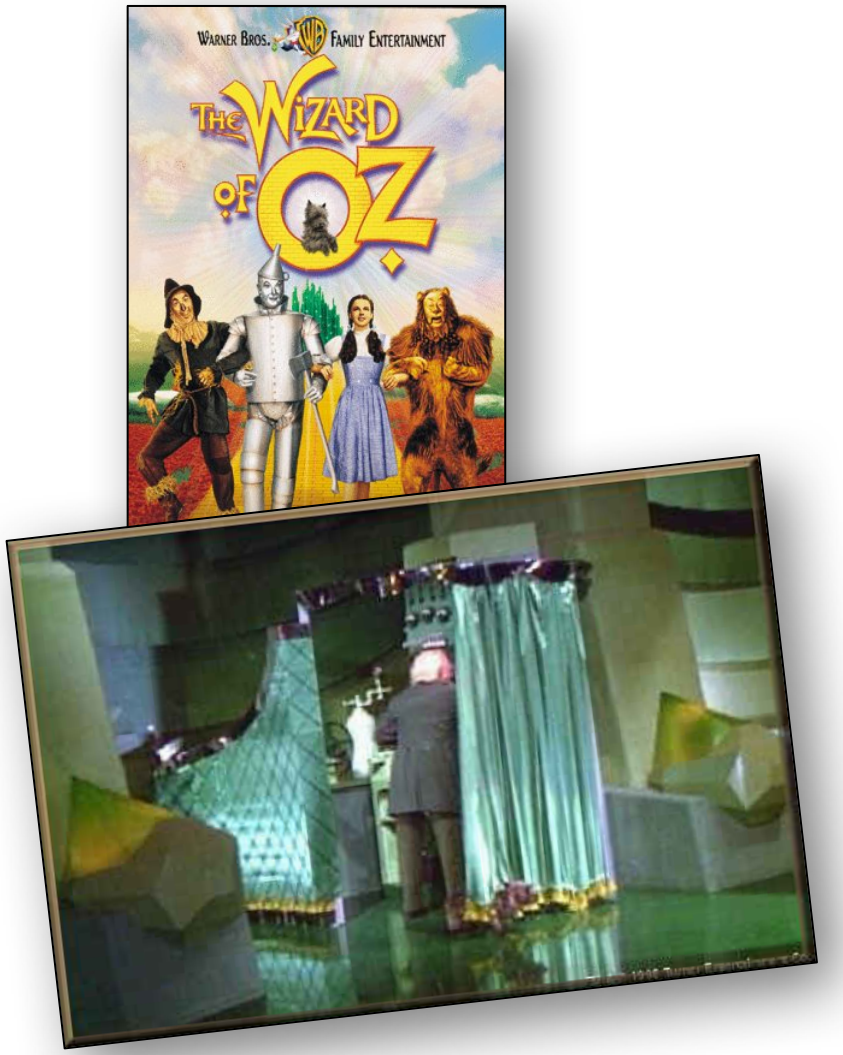
- Traditional storyboards
  - used if there is not a strong multimedia aspect
  - notes below each sketch contain the relevant steps from a *scenario*
  - the sketches themselves are annotated to indicate interactive behaviour
- Scored storyboards
  - if the application has a lot of motion graphics, the storyboard can be annotated
  - a sketch is annotated with appropriate notation and notes about, for example, type, colours, images, sound and other issues are attached underneath.
- Text-only storyboards
  - useful if the application has a lot of complex sequences


① INTRO SCENE
Butler
Newspaper
Competition
② Foyer
③ Foyer
Links to Rooms
④ Newspaper
⑤ Competition
Register
Lounge
⑥ Map
Guests
Book
Butler does room intro
Butler
③ Newspaper
③ Map
Introduce interface elements
Login Screen
"Still A Web Site"
⑦ Game
⑧ Simple overview
Wine
Conf
Pool
Billiards Room
⑨ Wine Cellar
Keypad
⑩ Show Newspaper + login/Register
Teaser
Full content.



The University Of Sheffield.

‘Wizard-of-Oz’ Prototyping



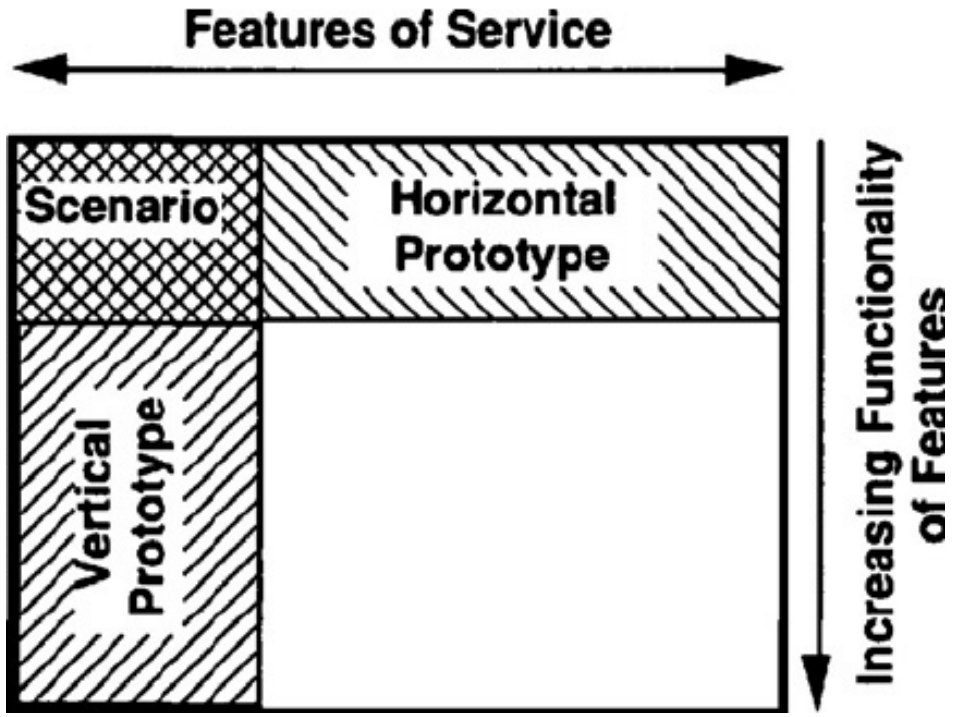
- The user thinks they are interacting with a computer, but a developer is responding to output rather than the system
- Usually done early in design to understand users’ expectations

Q: *what’s the big danger?*

A: *unrealisable human-level capabilities*

The University Of Sheffield.

Compromises in Prototyping



Nielsen

- All prototypes involve compromises ...
  - slow response
  - sketchy icons
  - limited functionality
- Two common types of compromise ...
  - 'horizontal': *provide a wide range of functions, but with little detail*
  - 'vertical': *provide a lot of detail for only a few functions*
- Compromises in prototypes mustn't be ignored (*the product needs engineering*)

The University Of Sheffield.

Pitfalls with 'Hi-Fi' Prototyping

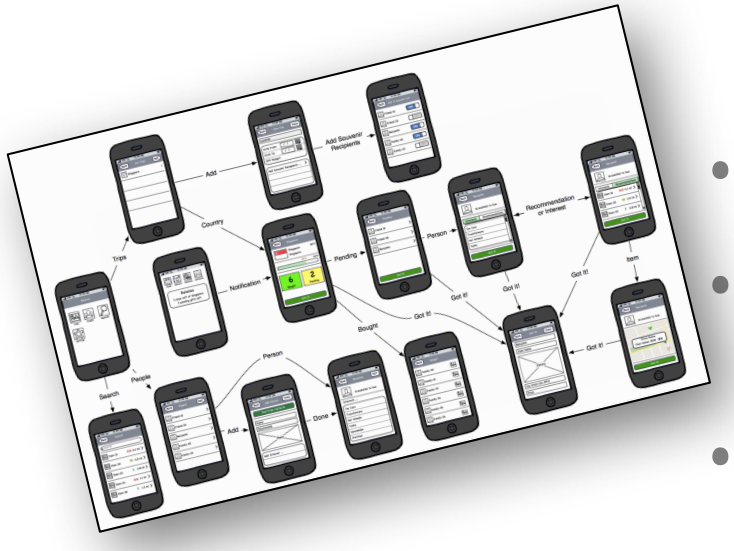
- People believe them!
  - dangerous if the designer has not checked details and thought the ideas through clearly beforehand
  - a simple error (*e.g. perhaps in the name of a customer, or of a product*) can completely ruin a prototype because clients or employees will get confused
- It suggests such a system can be implemented

The University Of Sheffield.

Pitfalls with 'Hi-Fi' Prototyping cont.

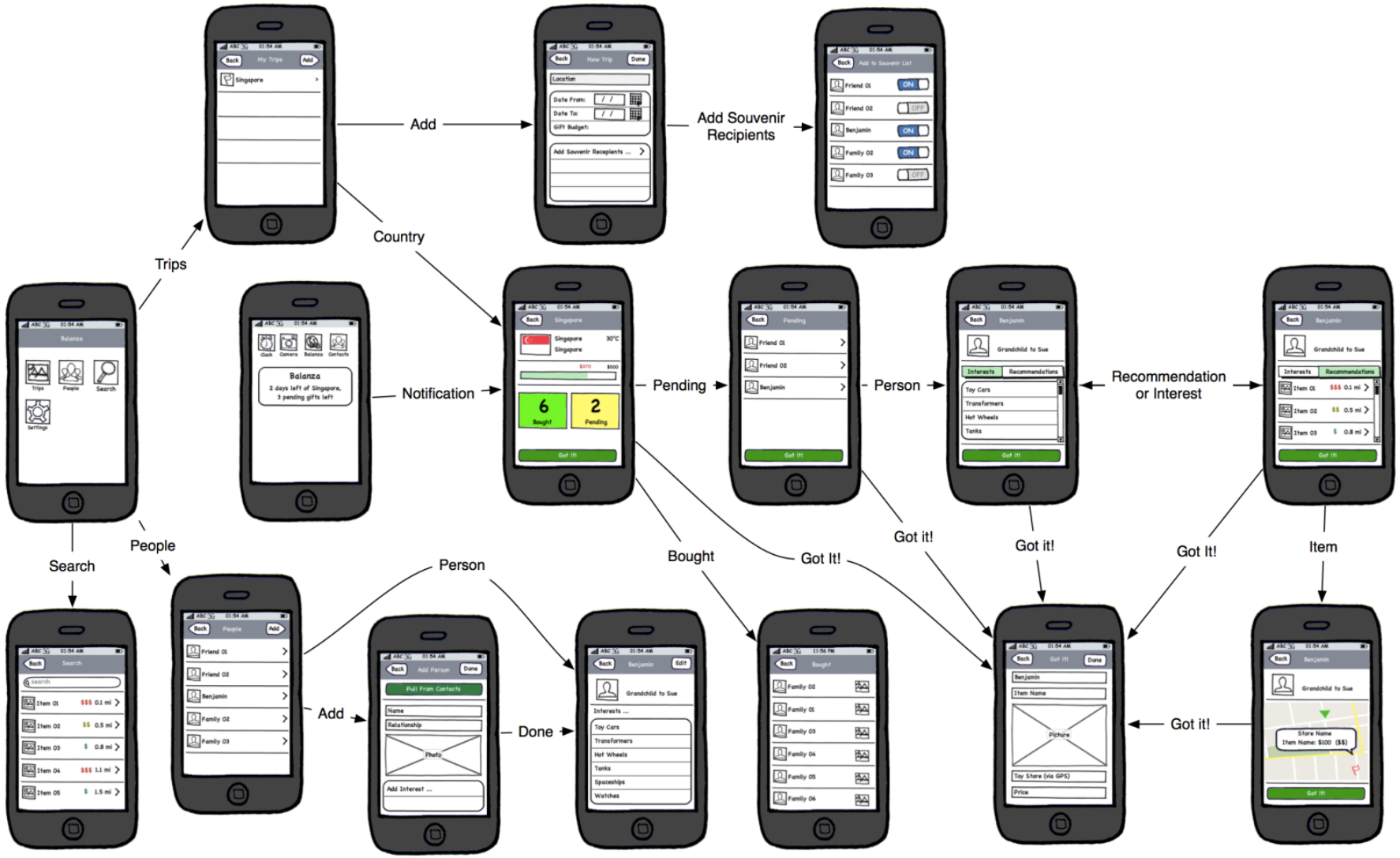
- A degree of effort and time is consumed in producing the prototype
- Developers can be reluctant to discard work on features rejected in exploring the prototype (*especially if the prototypes are created in the eventual development environment*)
- Users might be un-willing to genuinely criticise something that looks like it's taken a long time to develop
- Conclusion … *accurate* detail is vital!

Navigation Maps



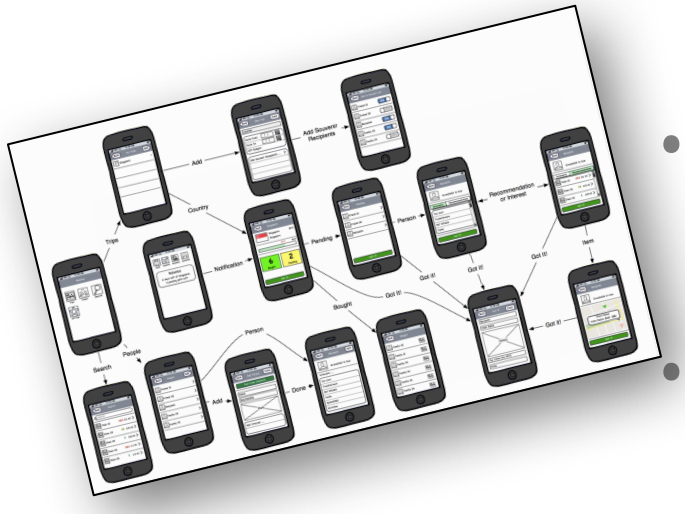
- Navigation is a key feature for many systems
- ‘Navigation maps’ focus on how people move through the site or application
- The aim is to focus on how people will experience the site/application
- Each page in the site, or location in the application, is represented with a box or heading and every page that can be accessed from that page should flow from it
- A useful tip is to put in all flows possible (*i.e. back and forwards from a page*) as this will highlight sections where people might get stranded

The University Of Sheffield.



http://www.jooyonglee.com/Balanza/navigation-map.html

Navigation Maps

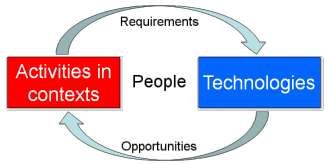


- Navigation maps can usefully be redrawn many times through the project life cycle (*poor navigational structure is one of the main reasons people turn off a website*)
- The maps can be used with scenarios to 'walk through' particular activities and are a very good way of spotting poor aspects of design such as 'orphan pages' (*pages which are not accessible*) or dead ends
- Navigation is important in all manner of applications and products, not just websites
- More formal and more fully annotated maps can also be developed
- Arrows can be added to lines if the direction of a link is important

The University Of Sheffield.

Trade-Offs in Prototyping

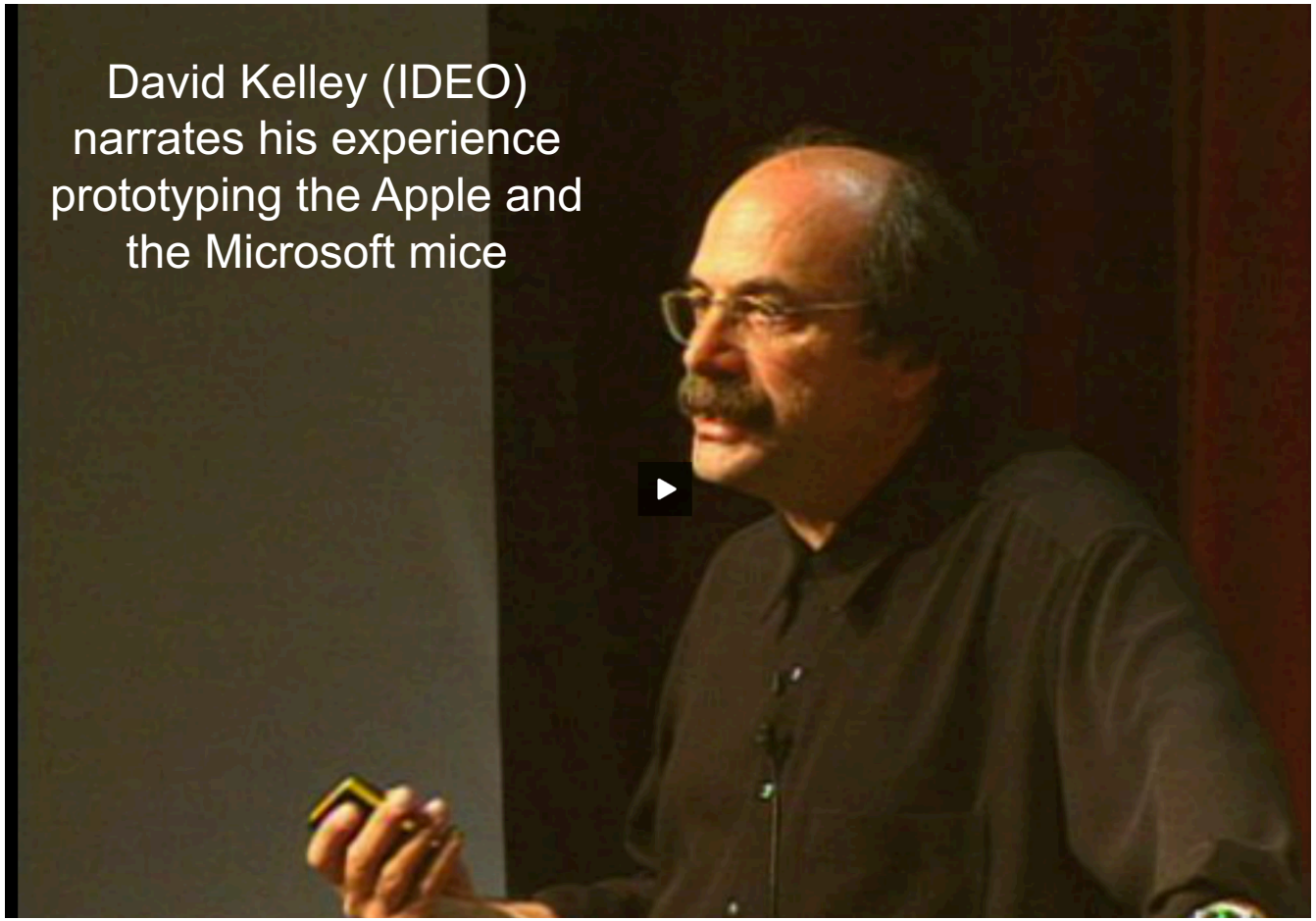
- The designer has to consider the trade-offs in terms of ...
  - time
  - resources
  - the aim of the evaluation
  - the stage of the project
  - etc.



- When reflecting on how and what to prototype, the designer should think in terms of the PACT elements (*people, activities, contexts and technologies*) ...
  - who is the prototype aimed at?
  - what is the designer trying to achieve with the prototype?
  - what stage of the project are things at and what is the context for the use of the prototype?
  - what technologies (*hi-fi or lo-fi*) are appropriate?

The University Of Sheffield.

Prototyping



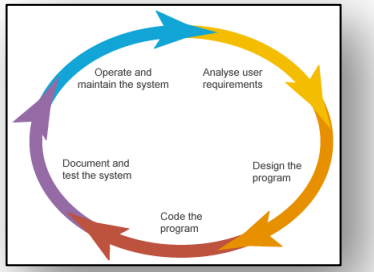
http://ecorner.stanford.edu/authorMaterialInfo.html?mid=687

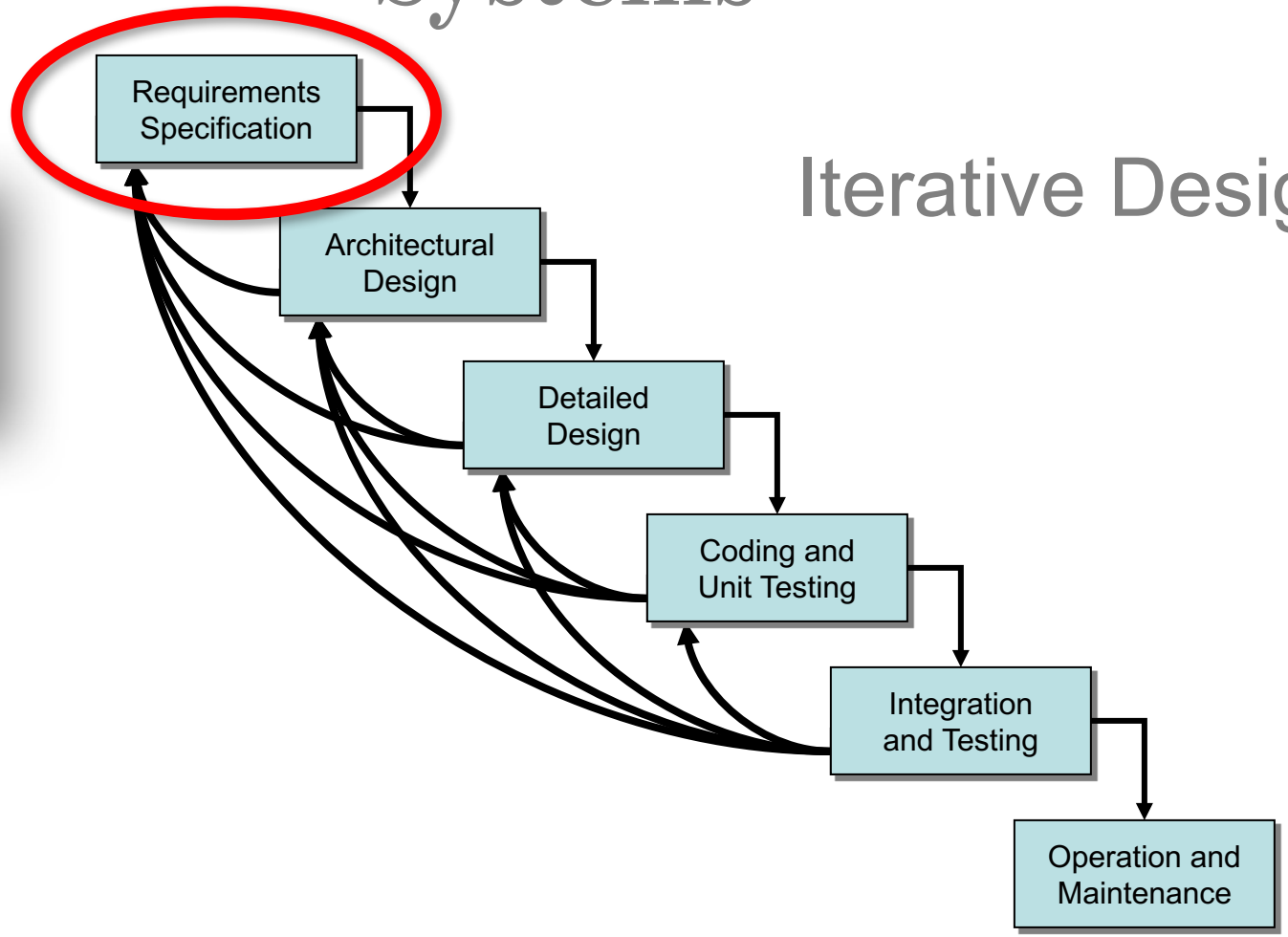
Exercise (*work in pairs*)

- Imagine that you are presenting ideas for a extreme sports goggles to a small team of developers from a manufacturer of tablet computers
- What type of prototype would you use (*and why*)?

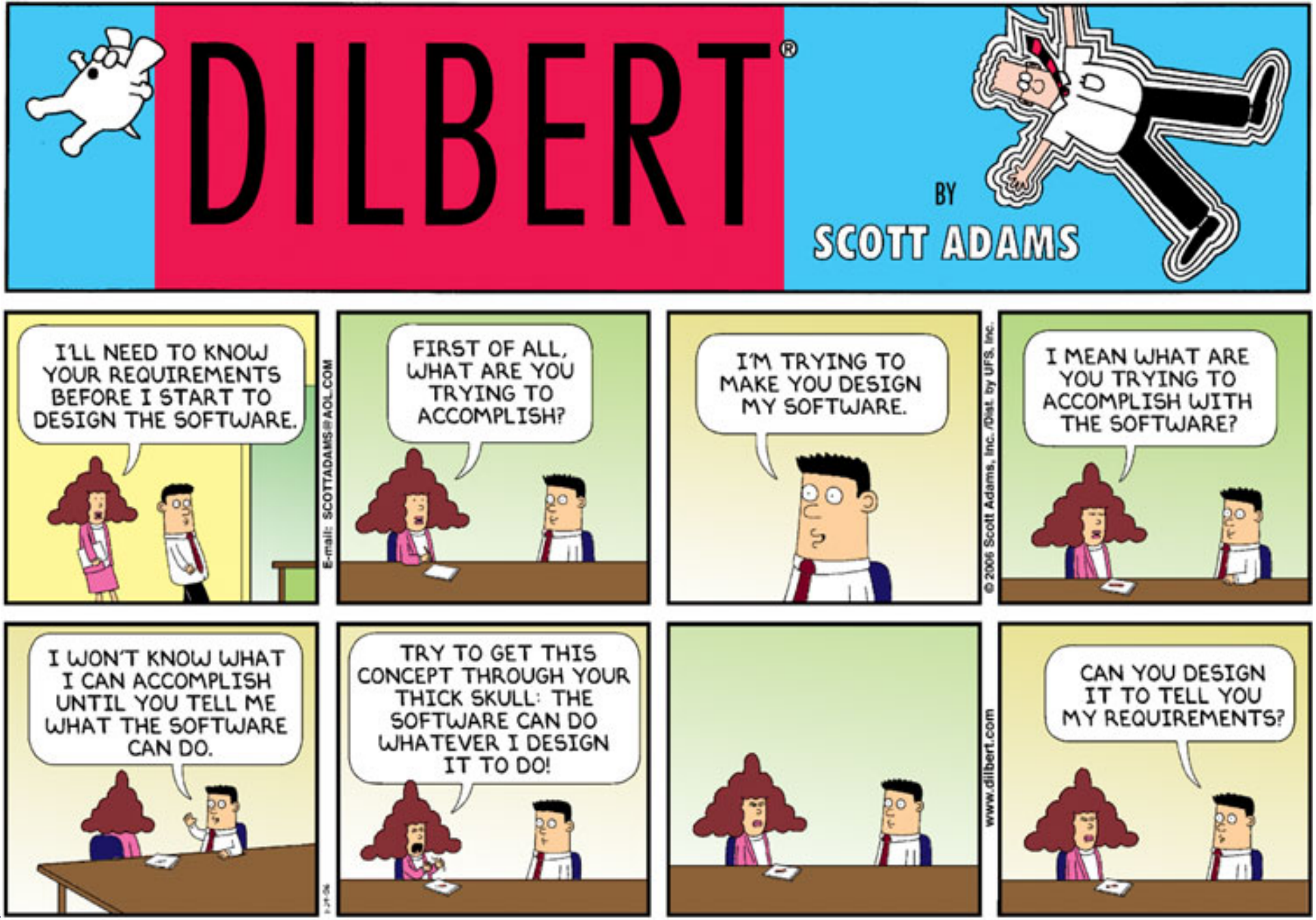
The University Of Sheffield.

The 'Life Cycle' for Interactive Systems





Iterative Design



‘Requirements’

“Something the product must do
or a quality that the product must have”

Robertson, S., & Robertson, J. (1999). *Mastering the Requirements Process*. Harlow, England: ACM Press/Addison-Wesley.

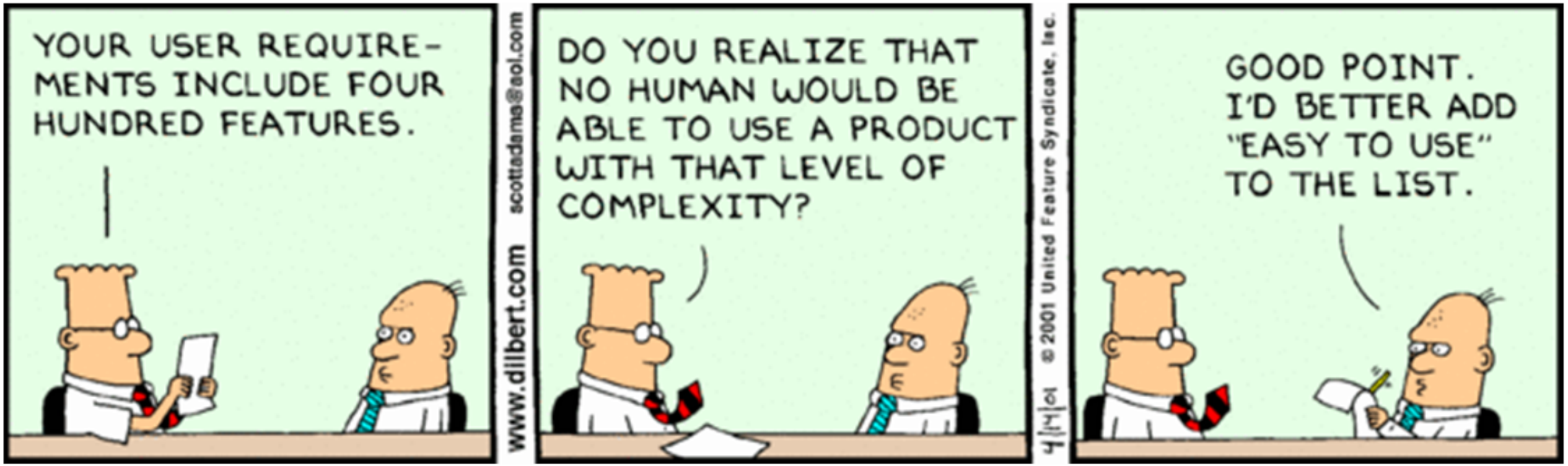
- Requirements are derived from information about …
  - the current situation
  - people’s needs, goals and aspirations
- Sometimes this is straightforward, but often it will need a creative leap (*this is why the process is so iterative*)
- The accuracy of the guess can only be judged when people review the requirements (*done with the aid of scenarios and early designs/prototypes*)
- Additional requirements will emerge as the design process continues.

The University Of Sheffield.

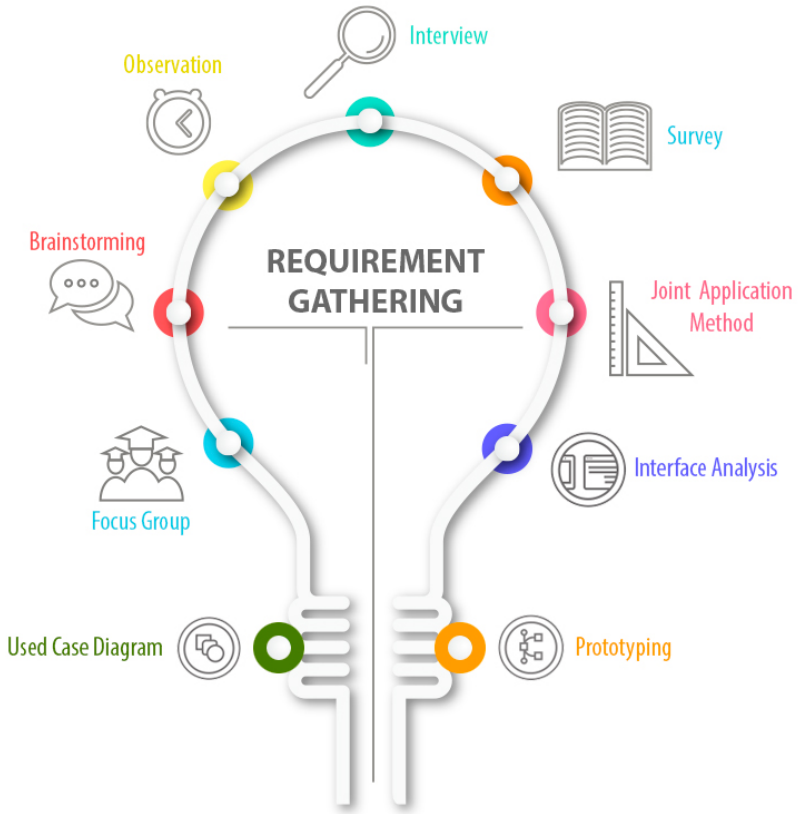
Requirements Specification

- Often clients will require a '**requirements specification**' (*a formal document that contains the requirements*)
- Developers also need a clear requirements specification at some point in the development process (*so that they can cost the project and manage it successfully*)
- Requirements specifications are often formal written documents, but increasingly they include ...
  - prototypes
  - screen shots
  - other media
- When written they should be ...
  - expressed in clear, unambiguous language
  - worded so that it will be possible to test whether the requirement has been met

Requirements Specification ....



Capturing Requirements



- Requirements gathering ...
  - suggests requirements are lying around waiting to be picked up with little interaction between designer and users
- Requirements generation ...
  - suggests a more creative activity
  - de-emphasizes links to current practice
- Requirements elicitation ...
  - suggests interaction between stakeholders and designers
- Requirements engineering ...
  - formal approach used in software engineering projects
- Requirements understanding ...
  - encapsulates gathering and generation

Capturing Requirements

- Requirements need to be identified within the *context* of use
- Need to take account of
  - 'stakeholders'
  - work groups and practices
  - organisational context
- Many approaches ...
  - socio-technical modelling
  - soft system modelling
  - participatory design
  - ethnography
  - contextual inquiry

Stakeholders

- A system will have many stakeholders
- A stakeholder is anyone effected by success or failure of system ...
  - 'primary' (*actually use system*)
  - 'secondary' (*receive output or provide input*)
  - 'tertiary' (*no direct involvement but effected by success or failure*)
  - 'facilitating' (*involved in development or deployment of system*)
- Designers need to meet as many stakeholder needs as possible
  - usually in conflict (*so have to prioritise*)
  - priority often decreases as move down categories

Stakeholders

Example: An airline booking system

An international airline is considering introducing a new booking system for use by associated travel agents to sell flights directly to the public.

Primary stakeholders:

travel agency staff, airline booking staff

Secondary stakeholders:

customers, airline management

Tertiary stakeholders:

competitors, civil aviation authorities,
customers' travelling companions, airline shareholders

Facilitating stakeholders:

design team, IT department staff

Socio-Technical Modelling

- Response to '**technological determinism**' (*early 20th century view that social change was dictated by technology*)
- The '**socio-technical systems**' view stressed that work systems were composed of human and machine elements
- Socio-technical models for interactive systems are concerned with technical, social, organisational and human aspects of design
- The process involves information gathering (*e.g. by interviews, observation, focus groups, document analysis, etc.*)

The University Of Sheffield.

Socio-Technical Modelling

- 'cUSToM': *User Skills and Task Match*
- Six stage process
  1. describe organisational context
  2. identify and describe stakeholders
  3. identify and describe work-groups
  4. identify and describe task–object pairs
  5. identify stakeholder needs
  6. consolidate and check stakeholder requirements against earlier criteria

Focused on stakeholders

Kirby, M. (1991). *Custom Manual* (DPO/STD/1.0): HCI Research Centre, University of Huddersfield.

The University Of Sheffield.

Socio-Technical Modelling

- 'OSTA': '*Open System Task Analysis*'

- Eight stage model
  1. primary task identified in terms of users' goals
  2. task inputs to system identified
  3. external environment into which the system will be introduced is described
  4. transformation processes within the system are described
  5. social system is analyzed
  6. technical system is described
  7. performance satisfaction criteria are established
  8. new technical system is specified

Focused on the task

Eason, K. D., & Harker, S. (1989). *An Open Systems Approach to Task Analysis*: HUSAT Research Centre, Loughborough University of Technology.

The University Of Sheffield.

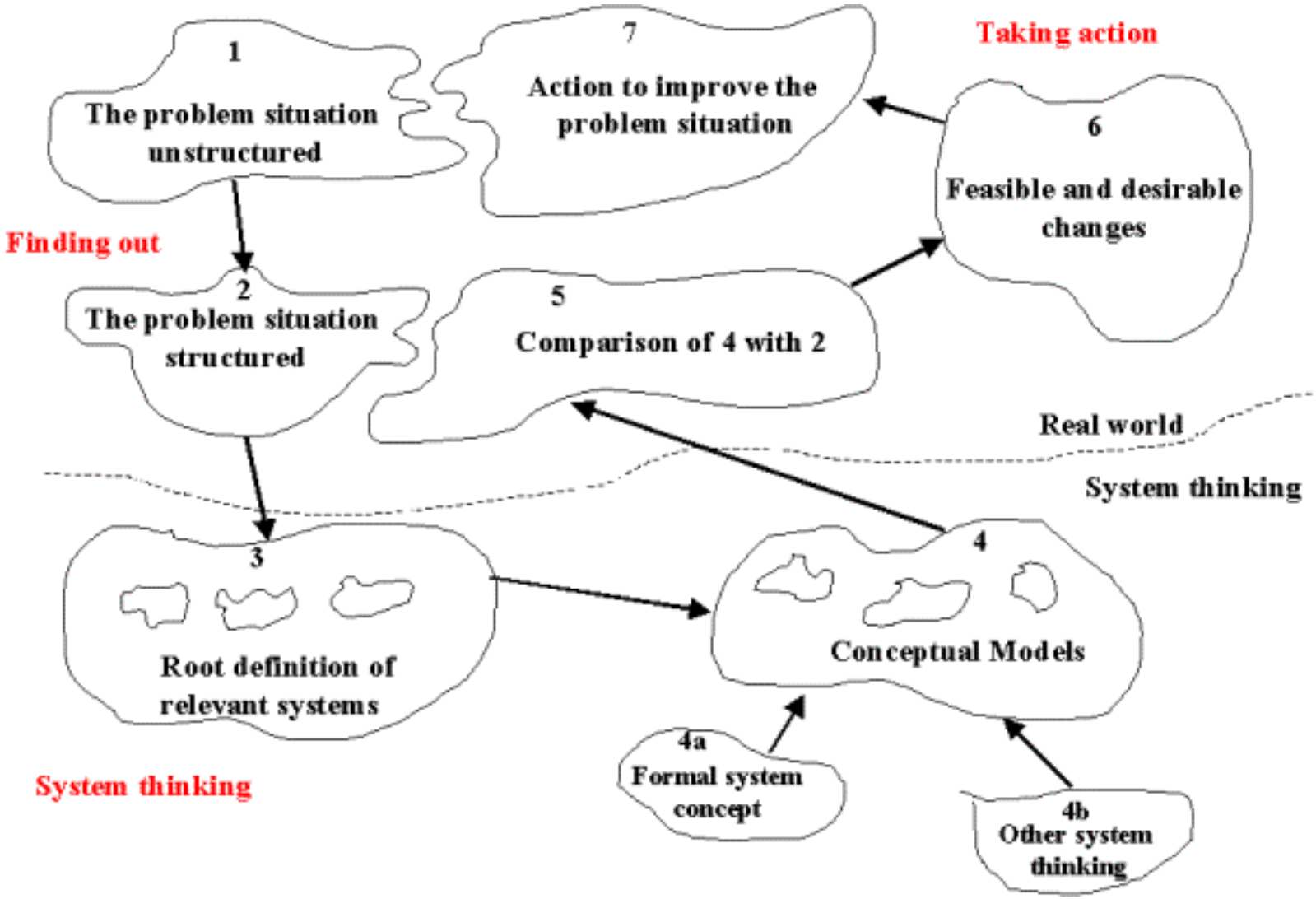
Socio-Technical Modelling

- 'Soft Systems Methodology' ...
  - no assumption of technological solution
  - emphasis on understanding situation fully
- Seven stages
  1. recognition of problem and initiation of analysis
  2. detailed description of problem situation
  3. generate root definitions of system ('*CATWOE*')
  4. conceptual model - identifying transformations
  5. compare real world to conceptual model
  6. identify necessary changes
  7. determine actions to effect changes

Checkland, P. B. (1981). *Systems thinking, Systems Practice*. Chichester: John Wiley.

The University Of Sheffield.

‘Soft Systems Methodology’



The University Of Sheffield.

CONGRESS

Anyway, we need to keep the roads clear for my Jaguar

Government is just expressing the will of the people

State Rep.

CITY HALL

influence

COURTS

Judge

Parking and pound recovery cases take up way too much time

influence

influence

influence

Lobbyist

This city is like a freeway – you can't move for cars. We need to keep cars out of the downtown area

Tax payer

Why do I pay so much tax to live in the city? You should raise tax from parking & road tolls!

THE CAR POUND

Funds

Funds

No special cases. Pay us $95/day or we keep the car

Pound Manager

Car owner

Pound admin

I'm sorry, but I don't have any record of your car being here.

My car's lost – the police told me it was towed here, but I can't find it.

Yeah, *sure* it was an emergency – I **hate** yuppies in sports cars

TRAFFIC & PARKING DIVISION

At least parking enforcement is a growth area – good pay & lots of jobs!

POLICE

Parking tickets

Sorry, but the ticket is issued – you can always challenge it in court

I'm sorry, but we just enforce the law – you need to speak to City Hall

Look, you can see how sick my baby is – I need to see the doctor

Why does everyone hate me? I'm just doing my job

That car has been there for five days – this used to be *such* a quiet neighborhood

D@^*! I can't park outside my house any more!

Local Businesses

We need City parking zones changed – it is too difficult to do business here

Driver

Exactly *how* does this relieve road congestion?

It's so dangerous to push a stroller into the road – why don't police ticket cars parked on the sidewalk?

Ow! I'm blind – how could I know there's a truck on the sidewalk?

The Univers Of Sheffiel

Socio-Technical Modelling

- 'Soft Systems Methodology' ...
  - no assumption of technological solution
  - emphasis on understanding situation fully
- Seven stages
  1. recognition of problem and initiation of analysis
  2. detailed description of problem situation
  3. generate root definitions of system ('*CATWOE*')
  4. conceptual model - identifying transformations
  5. compare real world to conceptual model
  6. identify necessary changes
  7. determine actions to effect changes

Checkland, P. B. (1981). *Systems thinking, Systems Practice*. Chichester: John Wiley.

The University Of Sheffield.

Socio-Technical Modelling

'CATWOE'

- Clients: *those who receive output or benefit from the system*
- Actors: *those who perform activities within the system*
- Transformations: *the changes that are affected by the system*
- Weltanschauung: *how the system is perceived in a particular root definition*
- Owner: *those to whom the system belongs, to whom it is answerable and who can authorize changes to it*
- Environment: *the world in which the system operates and by which it is influenced*

Socio-Technical Modelling

- In the '**Participatory Design**' approach, workers enter into the design context
  - user is an active member of the design team.
  - context and work oriented rather than system oriented
  - iterative and collaborative
  - employs brain-storming, storyboarding, workshops, pencil and paper exercises, etc.
- In the '**Ethnographic**' approach, designers enter into the work context
  - a form of anthropological study
  - designer does not enter actively into situation
  - seeks to understand social culture
  - unbiased and open ended
- Both approaches make workers feel valued in design (*and encourage them to 'own' the resulting products*)

The University Of Sheffield.

Participatory Design

- 'ETHICS': *Effective Technical and Human Implementation of Computer-based Systems*
  - system development is about managing change
  - non-participants more likely to be dissatisfied
- Three levels of participation
  - consultative
  - representative
  - consensus
- Design groups (*including stakeholder representatives*) make design decisions
- Job satisfaction is key to solution

Mumford, E. (1983). *Designing Participatively*. Manchester: Manchester Business School Publications.

Participatory Design

http://www.youtube.com/watch?v=4npftEf3_n4

Ethnographic Approach

- 'Contextual Inquiry'
  - investigator apprenticed to user to learn about work
- Investigation takes place in the workplace using ...
  - detailed interviews
  - observation
  - analysis of communications
  - physical workplace
  - artefacts
- Several models are created, and output indicates ...
  - task sequences
  - artefacts and communication channels needed
  - physical and cultural constraints

Beyer, H., & Holtzblatt, K. (1998). *Contextual Design: Defining Customer Centered Systems*: Morgan Kaufmann.

Task Analysis

- The process of analysing the way in which people perform their jobs
  - the things they do
  - the things they act on
  - the things they need to know
- Task analysis is not limited to activities including a computer
- Task analysis is focused on the user
- The most popular technique is '**hierarchical task analysis/decomposition**' (*HTA*)

The University Of Sheffield.

Hierarchical Task Analysis

- Tasks are broken down into 'sub-tasks', then 'sub-sub-tasks' and so on
- Tasks and sub-tasks are grouped as 'plans' which specify how the tasks might be performed in practice
- HTA focuses on physical and observable actions, and includes looking at actions not related to software or an interaction device
- Process
  - start with the user goal
  - identify the main tasks for achieving it
  - decompose tasks into sub-tasks, sub-sub-tasks, etc.
  - only decompose relevant (*sub*)tasks

Hierarchical Task Analysis

Example 'task hierarchy' ...

0. Clean the house
    1. get the vacuum cleaner out
    2. get the appropriate attachment
    3. clean the rooms
        3.1 clean the hall
        3.2 clean the living rooms
        3.3 clean the bedrooms
    4. empty the dust bag
    5. put vacuum cleaner and attachments away

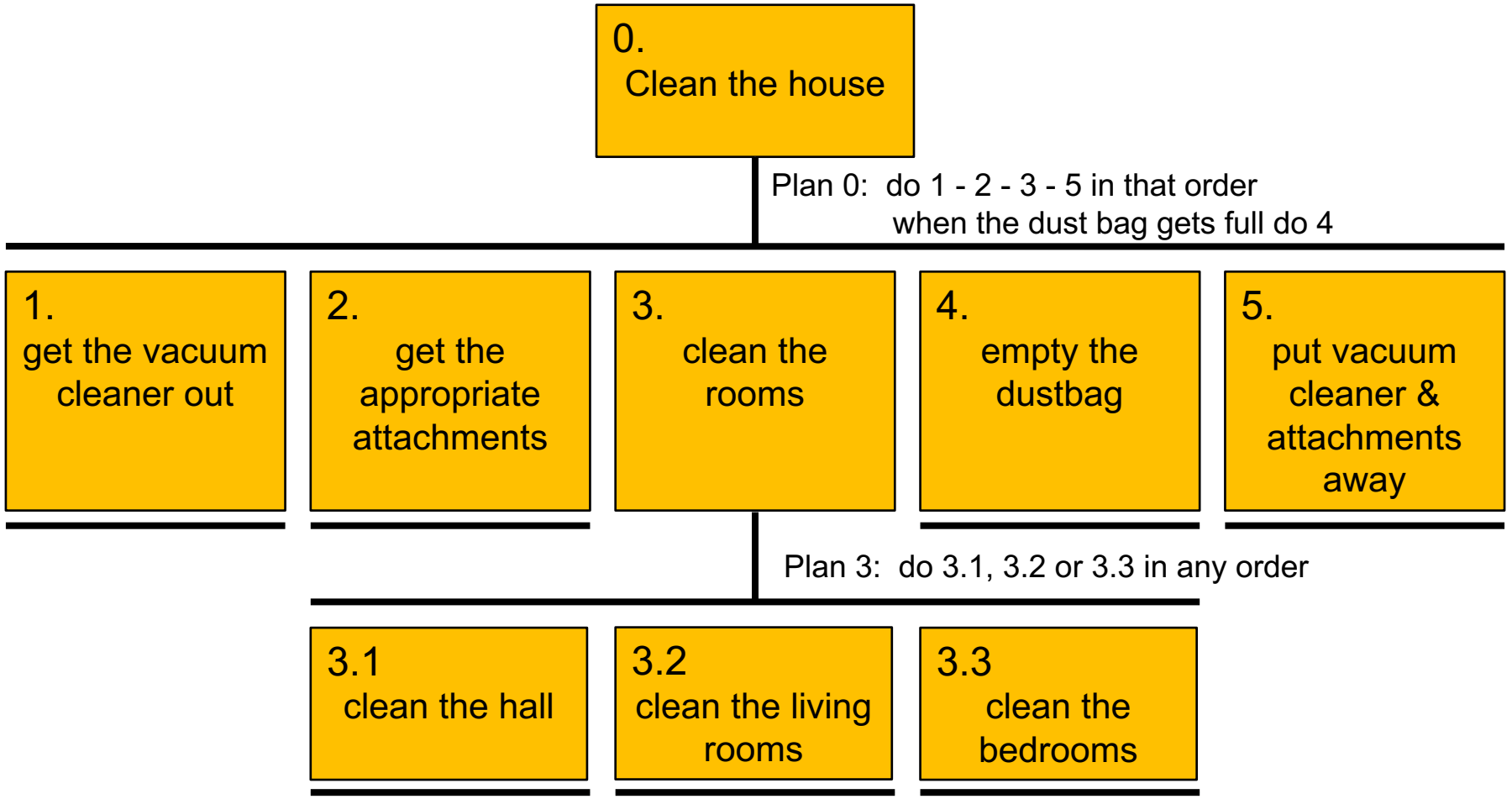
Plan 0: *do 1 - 2 - 3 - 5 in that order*
when the dust bag gets full do 4

Plan 3: *do 3.1, 3.2 or 3.3 in any order*

Only the *plans* specify an order

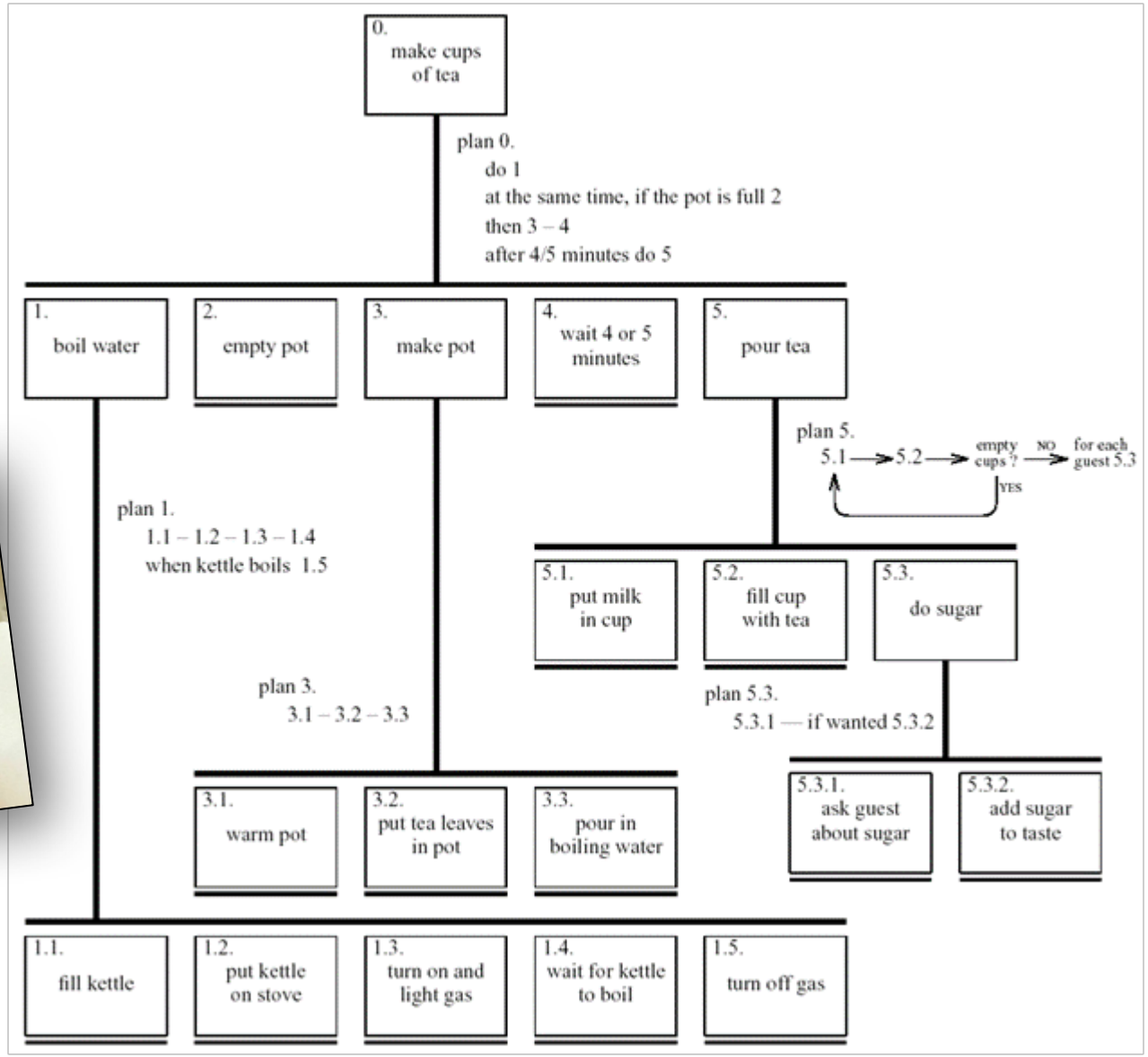
The University Of Sheffield.

Hierarchical Task Analysis



Task Analysis

http://www.youtube.com/watch?v=1KML29WefLo


0.
make cups of tea
plan 0.
do 1
at the same time, if the pot is full 2
then 3 – 4
after 4/5 minutes do 5
1.
boil water
2.
empty pot
3.
make pot
4.
wait 4 or 5 minutes
5.
pour tea
plan 5.
5.1 → 5.2 → empty cups? → NO → for each guest 5.3
YES
plan 1.
1.1 – 1.2 – 1.3 – 1.4
when kettle boils 1.5
5.1.
put milk in cup
5.2.
fill cup with tea
5.3.
do sugar
plan 3.
3.1 – 3.2 – 3.3
plan 5.3.
5.3.1 — if wanted 5.3.2
3.1.
warm pot
3.2.
put tea leaves in pot
3.3.
pour in boiling water
5.3.1.
ask guest about sugar
5.3.2.
add sugar to taste
1.1.
fill kettle
1.2.
put kettle on stove
1.3.
turn on and light gas
1.4.
wait for kettle to boil
1.5.
turn off gas

Structural Knowledge

- Task analysis is about 'procedural knowledge'
- However, before a person sets about some procedure they need to know what types of things can be accomplished in a domain
- E.g. a user of a drawing package needs to know that there is a facility for changing the thickness of a line, before working out how to do it
- Users need some conception of what is possible, i.e. 'structural knowledge'
- An appropriate technique for capturing structural knowledge is 'entity-relationship modelling'

Entity-Relationship Modelling

- Technique more usually associated with ...
  - data-base design
  - object-oriented programming
- The difference is that entity-relationship modelling includes non-computer entities ...
  - physical objects
  - actions performed on them
  - the people who perform the actions

Example

Design a new computer controlled irrigation system for a market gardening firm (‘*Vera's Veggies*’)

Owner/manager: *Vera Bradshaw*
Employees: *Sam Gummage and Tony Peagreen*
Assets: *various tools including a tractor (`Fergie‘), two fields and a glasshouse*

The University Of Sheffield.

Entity-Relationship Modelling

- Start with list of 'objects' and classify them ...
  - concrete objects:
    - simple things: *spade, plough, glasshouse*
  - actors:
    - human actors: *Vera, Sam, Tony, the customers*
    - the irrigation controller
  - composite objects:
    - sets: *the team (Vera, Sam, Tony)*
    - tuples: *the tractor (Fergie + plough)*

- Add 'attributes' to the objects

Object Pump3 **simple** – irrigation pump
Attributes:
status: on/off/faulty
capacity: 100 litres/minute

Entity-Relationship Modelling

- List 'actions' and associate with each ...
  - 'agent ' (*who performs the actions*)
  - 'patient' (*which is changed by the action*)
  - 'instrument' (*used to perform action*)

"Sam (*agent*) planted (*action*) the leeks (*patient*)."
"Tony (*agent*) dug (*action*) the field (*patient*) with the spade (*instrument*)."

- Issues
  - implicit agents (*read behind the words*)
    "*The field was ploughed*" – by whom?
  - indirect agency (*the real agent?*)
    "*Vera programmed the controller to irrigate the field*"
  - Messages (*a special sort of action*)
    "*Vera told Sam to ...* "
  - Rôles (*an agent acts in several rôles*)
    Vera as worker or as manager

The University Of Sheffield.

Entity-Relationship Modelling

Object Sam **human actor**
Actions:
S1: drive tractor
S2: dig the carrots

Object Vera **human actor**
– the proprietor
Actions: as worker
V1: plant marrow seed
V2: program irrigation controller
Actions: as manager
V3: tell Sam to dig the carrots

Object the men **composite**
Comprises: Sam, Tony

Object glasshouse **simple**
Attribute:
humidity: 0-100%

Object Irrigation Controller
non-human actor
Actions:
IC1: turn on Pump1
IC2: turn on Pump2
IC3: turn on Pump3

Object Marrow **simple**
Actions:
M1: germinate
M2: grow

Entity-Relationship Modelling

- 'Events' ...
  - performance of action
    "Sam dug the carrots"
  - spontaneous events
    "the marrow seed germinated"
    "the humidity drops below 25%"
  - timed events
    "at midnight the controller turns on"

- 'Relationships'
  - object-object
    - social (*Sam is subordinate to Vera*)
    - spatial (*pump 3 is in the glasshouse*)
  - action-object
    - agent (*listed with object*)
    - patient and instrument
  - actions and events
    - temporal and causal
      "Sam digs the carrots because Vera told him"
  - temporal relations
    - use HTA

Entity-Relationship Modelling

Events:
Ev1: humidity drops below 25%
Ev2: midnight

Relations: object-object
location (Pump3, glasshouse)
location (Pump1, Parker's Patch)

Relations: action-object
patient (V3, Sam)
– Vera tells *Sam* to dig
patient (S2, the carrots)
– Sam digs the *carrots* ...
instrument (S2, spade)
– ... *with* the spade

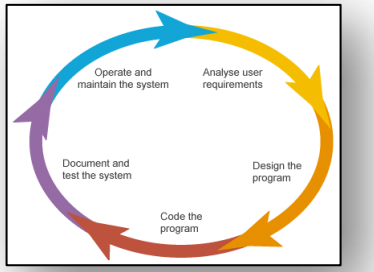
Relations: action-event

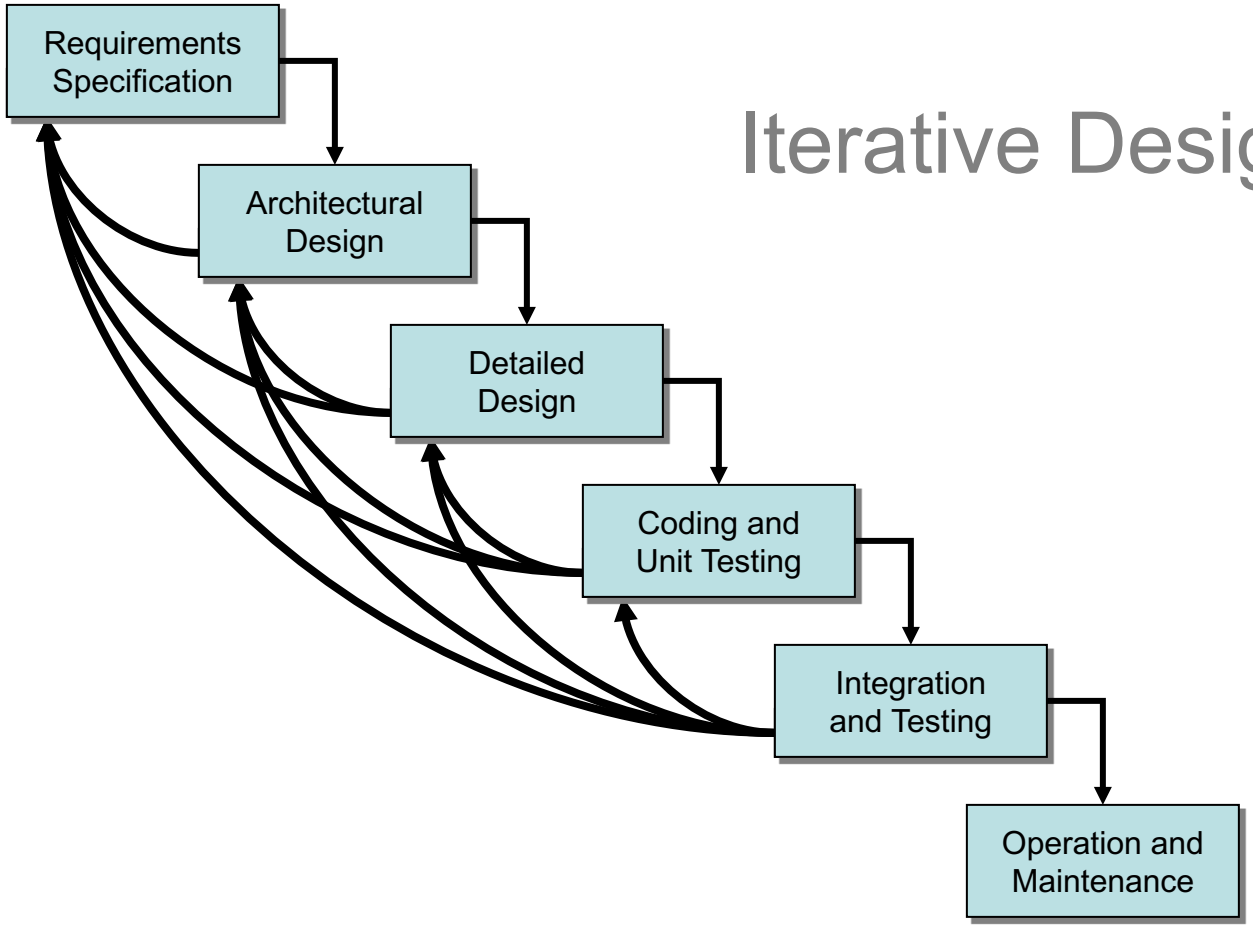
before (V1, M1)
– the marrow must be sown *before* it can germinate

triggers (Ev1, IC3)
– *when* humidity drops below 25%, the controller turns on pump 3

causes (V2, IC1)
– the controller turns on the pump *because* Vera programmed it

The 'Life Cycle' for Interactive Systems

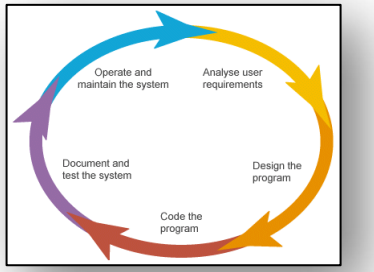




Iterative Design

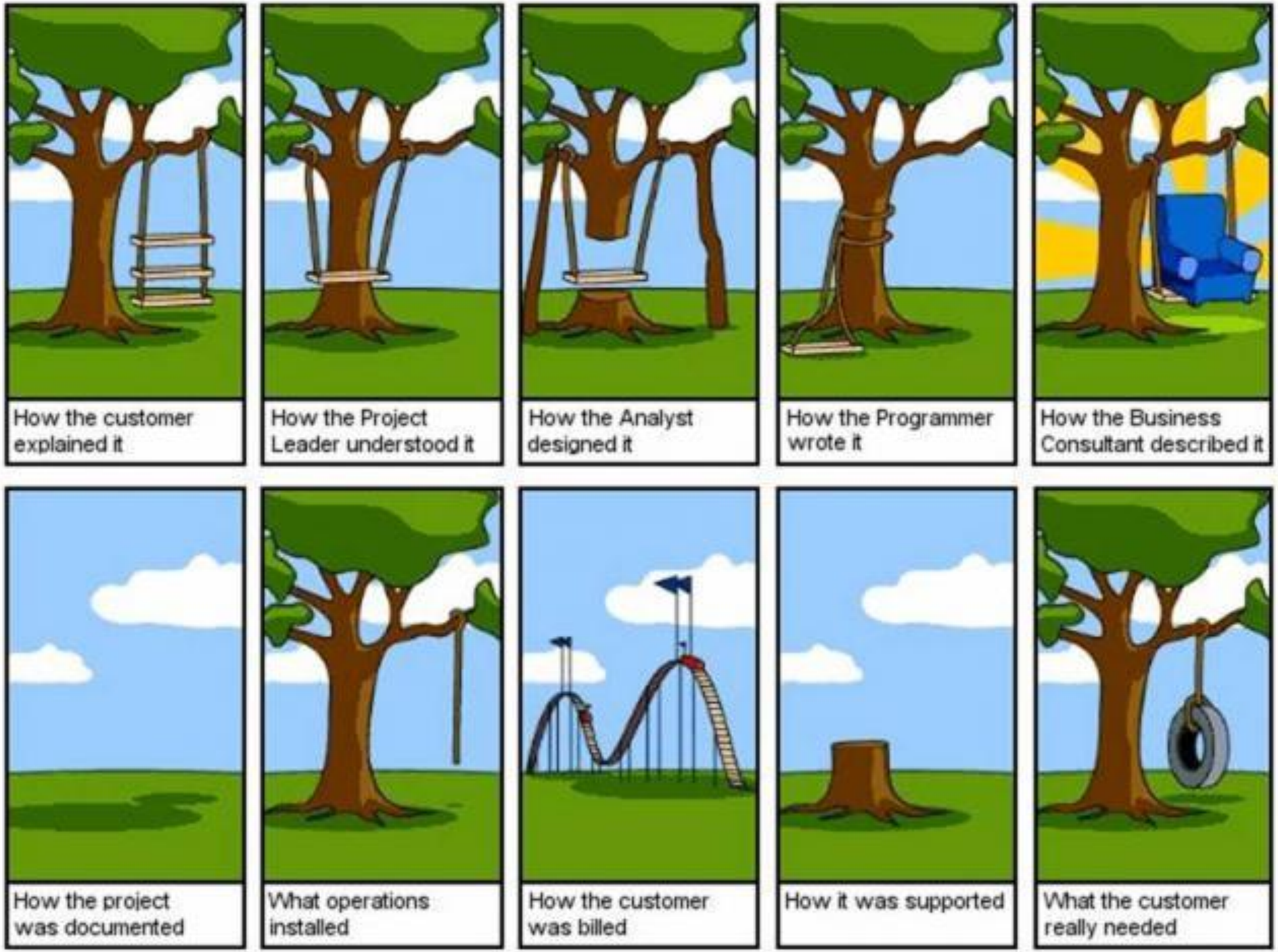
The University Of Sheffield.

Downside of Iterative Design



- Management issues ...
  - time
  - planning
  - non-functional features
  - contracts
- Design inertia ...
  - early bad decisions stay bad
- Diagnosing usability problems ...
  - important to understand the reason for a problem (*not just the symptoms*)

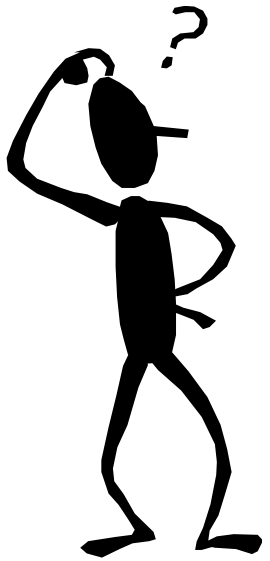
The University Of Sheffield.


How the customer explained it
How the Project Leader understood it
How the Analyst designed it
How the Programmer wrote it
How the Business Consultant described it
How the project was documented
What operations installed
How the customer was billed
How it was supported
What the customer really needed

This lecture has covered …

- The software life cycle
- Iterative design
- Prototyping
- Storyboarding
- Navigation maps
- Requirements capture
- Socio-technical modelling
- Hierarchical task analysis

Any Questions or Feedback ?



Thank You

Enjoy the rest of the course ☺

The University Of Sheffield.

Next time …

Measuring Usability