MLMI10 Designing Intelligent Interactive Systems
Lecture 3

Per Ola Kristensson

Lent 2022

Key Concepts

- **Target audience**
  - Who are we designing for?
- **Task clarification**
  - How do we address the *right* problem by evolving a solution-neutral problem statement?
  - How do we specify requirements that are both valid, relevant and verifiable?
- **Understanding users**
  - What are the users needs and wants?

Why do products fail? Some reasons…

- Solving the wrong problem: unable to identify the correct target audience and/or understand user needs
- Solution in search of a problem: no market-pull factors; all technology-push
- Works in the lab but cannot be manufactured: poor product architecture, reuse, understanding of manufacturing processes and constraints
- Unable to elicit correct requirements, poor requirements engineering process
- Elicited misleading requirements: common error: believe user speculation of what they want to buy, however, most users are unable to predict their own behaviour
- Too incremental: outcompeted; race to the bottom won by whoever has succeeded in logistics
- Too far ahead on the curve: the market is not there yet
- Poor performance
- Too expensive for target audience
- Poor trade-off between expected eventual performance and learning effort demanded by users
- **Failure to identify all critical design parameters: trade-offs made implicitly in the design rather explicitly as part of an informed design process**

Design engineering

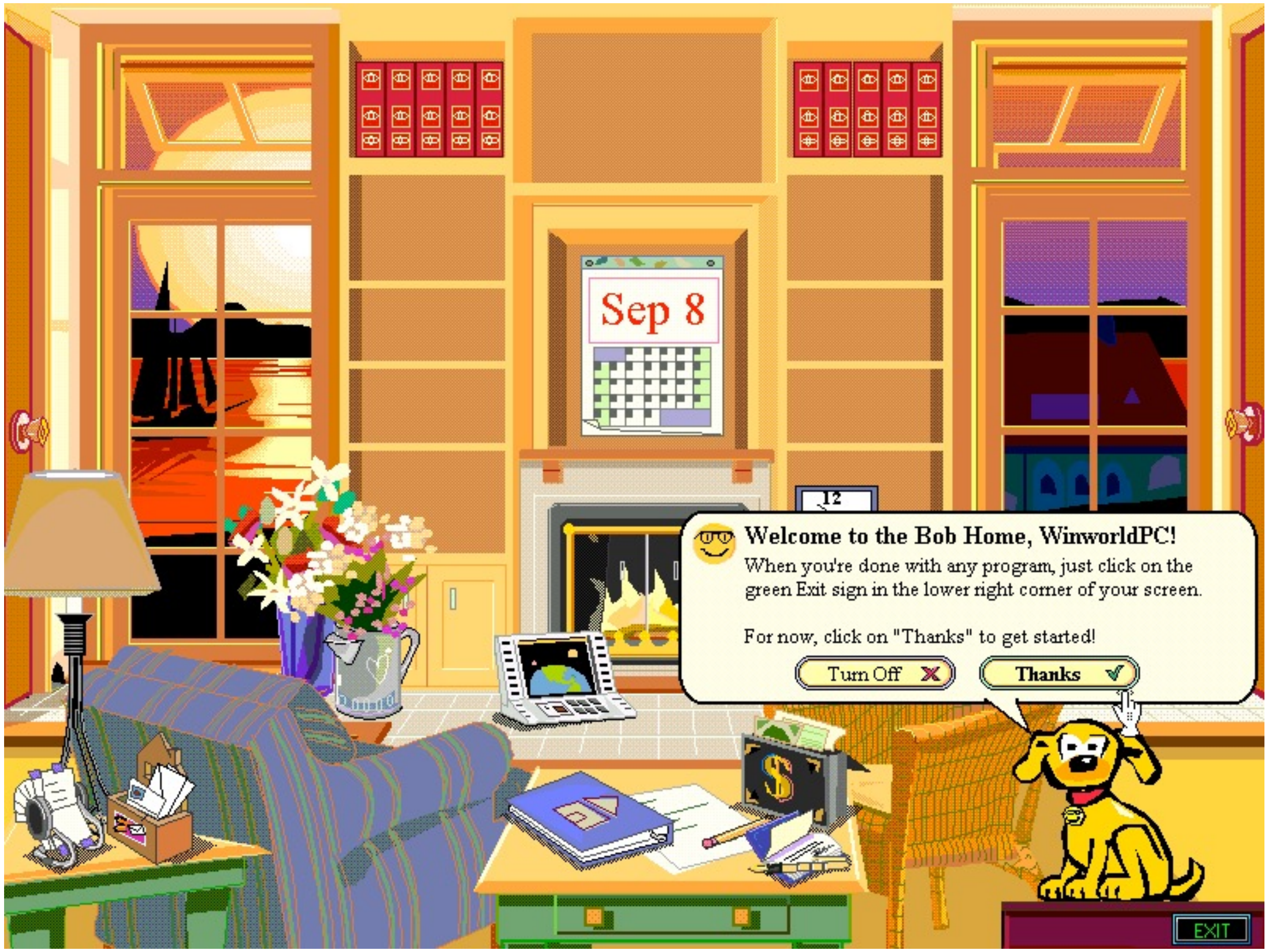
- **Design engineering** is a set of methodologies used in engineering to design products, services and systems
- The overall process is as follows:
  - Task clarification
  - Conceptual design
  - Embodiment design
  - Detailed design
  - Verification and validation
- Design engineering is suitable for developing intelligent interactive systems as successful deployment of such systems necessitate intimate knowledge of all functions and their associated controllable and uncontrollable parameters

Solving the right and wrong problems

- A very common mistake is to solve the **wrong** problem
- There are many reasons why this is so prevalent; usually the source is poor understanding of the task itself, for instance due to poor requirements elicitation or procurement (often a combination of both)

Wrong problem: exhibit 1

- Microsoft Bob
- Windows Shell for novices
- Perceived problem: novice users need a physical world metaphor to understand computing



Welcome to the Bob Home, WinworldPC!
When you're done with any program, just click on the green Exit sign in the lower right corner of your screen.

For now, click on "Thanks" to get started!

https://winworldpc.com/res/img/screenshots/100-81b432cbcb492b953f42417b8cf5a95c-Microsoft%20BOB%201.0%20-%20Main.png

Wrong problem: exhibit 2

- Typing with a stylus on a resistive screen was tedious and error prone
- Perceived problem 1: lack of natural input: hand printing or handwriting recognition
- Perceived problem 2: Unoptimised QWERTY keyboard design lends itself to the pen zig-zagging between the left and right hand sides of the keyboard

Identifying the right problem

- To succeed in identifying the right problem we must first gain an in-depth understanding of the following:

1. The **target audience**
2. Users' **needs and wants**

- This allows us to provide a narrative of the **problem context** that motivates the design effort
- From the problem context we can construct a **problem statement** and evolve it into a **solution-neutral problem statement**
- This process avoids solutions in search of problems and solutions that result in a poor fit with users' desires and expectations

Target Audience

Understanding the target audience

- The **target audience** is a description of the intended customers/users and other stakeholders affected by your device
- Understanding the target audience is **critical** for ensuring the right problem is solved in the right way
- A major design risk is failing to understand the target audience, either due to an inability to describe the audience (called a **sampling frame**) or due to an inability to sample representatively from it (called **sampling error**)

Pick your audience

- The longer a company has been around, the more likely the company knows its audience (customers and/or procurers)
  - Detailed information about customers (and end-users) might be available and could be used for recruiting
- Early-stage research to discover business opportunities require first identifying who the users could be
  - You should define your audience and recruit those people in an initial round of research to confirm your assumptions
  - If those people are not interested in your wearable device you might need to recruit different people for another round

Purchasers vs. end-users

- It is important to question assumptions about audiences
- Companies used to traditional marketing research often want purchasers or final decision markers (such as general managers) as participants
- The reality is often that purchases and final decision makers do not use the products or services they are responsible to purchase
- If you recruit them (based on for example client requests) there is a risk that you are designing for an audience that do not use your product or service
- This is a very common phenomenon in industry and often a source of pain for end-users stuck with the product or service
- In such a situation you might argue for **two** studies, usability research with end-users and surveys and/or interviews with decision makers about purchasing behaviours

Defining “right people” for an audience

- The **behavioural** criterion:
  - You want people who actually do (or want to do) the things your intelligent interactive system supports
- The **technological** criterion:
  - For some intelligent interactive systems it is important to know how intensively they use (or don’t use) certain computing and communication technology
- The **demographical** criterion:
  - Age, gender, geographical location, household income, etc.
  - For new products or services this criterion might unnecessarily restrict your recruitment pool
  - For established products and services it can be important that participants model your existing demographical profile of your existing customer (end-user) base

Profiles of target audience, initial version

- Example: Wearable health monitor coupled with smartphone to use persuasive gamification methods to encourage a healthy life style

Demographics

- Ages: 25-55
- Gender: male or female
- Single
- Educated at university-level
- Income: £30K+

Behaviours

- Walks / uses the tube to go to work

Technology Use and Experience

- Use their mobile phone to browse the web and Facebook several times per day

Revise the profile

- Ask yourself what makes the ideal participants different from the target audience as a whole
  - What kind of people will give the best feedback for the specific research you are doing?
- Ask yourself:
  - Which *segments* of your audience do you need to focus on?
  - How much experience should they have with *your product?* (Do you want fresh users?)
  - How much experience should they have with *competing products?*
  - Are you targeting a single group of users or multiple groups? (For example: city-dwellers and suburbanities might be two groups)
  - What are undesirable *characteristics* that should be avoided? (For example: people who consider themselves experts on their cities might not provide useful feedback)
- Explore answers to these questions and revise the profile
- Remove factors that are not going to affect how people use or view the product and add information
- Focus on isolating the factors that will make them an ideal audience

Profiles of target audience, initial version: revised

- Example: Wearable health monitor coupled with smartphone to use persuasive gamification methods to encourage a healthy life style

Demographics

- Ages: 25-55
- Gender: male or female
- Single
- Educated at university-level
- Income: £30K+

Behaviours

- Walks / uses the tube to go to work
- Prefer not to exercise (gym, running, etc.)
- Does not wear a (normal) watch
- Talks about health issues with friends

Technology Use and Experience

- Use their mobile phone to browse the web and Facebook several times per day

The “average user” and multiple profiles

- The “average user” is not always an appropriate model
  - If you are looking for new ideas to improve or differentiate a product or service, people who use it intensively, have an unusual way of doing it or have special needs might provide more useful input
  - Sometimes it is even useful to interview people who have not used the product or service
- It is possible to split a profile into multiple profiles
- A profile should not be over-determined
- If you think criteria in your profile makes it unlikely for you to recruit participants or some restrictions are mutually exclusive then split the profile into multiple profiles

Personas

- Understanding the target audience enables the construction of **personas**
- Personas are fictional characters created to model prototypical users
- Often created by synthesizing data from surveys or interviews
- Advantages of personas:
    1. Personas put a human face to an otherwise abstract “customer”
    2. Personas help team members sharing a consistent understanding of who the end-users are supposed to be
    3. The benefit of solutions and specific features can be understood by directly relating them to the personas

Task Clarification

The importance of task clarification

- To avoid solving the wrong problems it is essential to understand how to build the right thing
  - Addressed by evolving a solution-neutral problem statement and understanding users’ and other stakeholders’ needs and wants
- In addition, having decided to build the right thing, it is of course important we build the thing right
  - Addressed by evolving a requirements specification

Clarify the task

- There are two important steps in the task clarification phase of the design process:

1. Preparing a problem statement
2. Elaborating a specification

Solution-neutral problem-statement

- To avoid solving the **wrong** problem, it is wise to spend some time identifying the true needs and preparing a **solution-neutral problem statement** which avoids any indication of how the problem should be solved. A useful technique is to systematically raise the level of abstraction.

Example

Consider the problem statement:

*"Design a movie recommendation system to replace the system deployed **last year**. It must be capable of handling at least 100,000 movies, **use a deep neural network architecture**, and explain recommendations to the user."*

This statement clearly indicates the direction of the solution.

A solution-neutral problem statement may be derived by successive abstraction:

- Replace **last year's system**
- Design a **movie recommendation system**
- Create a system that helps users locate **movies**
- Devise a means of enticing and engaging users in movies
- Plan a way of entertaining users

increasing abstraction

Abstraction

- Abstraction has the following steps:
  - Eliminate requirements which have no direct bearing on the function and essential constraints
  - Transform quantitative statements into qualitative ones
  - Formulate the problem in solution-neutral terms at the appropriate level of generality

Abstraction

- Abstraction has the following steps:
  - Eliminate requirements which have no direct bearing on the function and essential constraints
  - Transform quantitative statements into qualitative ones
  - Formulate the problem in solution-neutral terms at the appropriate level of generality
- **Abstraction increases the search space**

Requirements

Requirements specification

- A **requirements specification** is determined in the beginning of the design process
- It will be frequently referred to later in the design process
  - Therefore it is very important it is correct
  - Requirements specifications are therefore reviewed for correctness
- A good requirements specification:
  - provides an in-depth understanding of the problem
  - decreases time (and cost) in the design process
  - improves communication among team members
  - is more likely to result in a quality product
  - makes it easier to make an intelligent interactive system compliant with various regulations and standards
- Requirements specification in tandem with user-centred design is referred to as **requirements engineering**

Requirements

- A requirements specification translates needs into engineering terms
- There are different types of requirements:
  - User-elicited requirements
  - Technical requirements
  - Business requirements
  - Regulatory requirements
- Characteristics of effective requirements:
  - Solution independent
  - Complete
  - Clear
  - Concise
  - Testable
  - Traceable

Requirements specification process

- Phase 1
  - Problem definition and definition of objectives
- Phase 2
  - Detailing of functions that must be performed to satisfy the problem(s) defined in Phase 1
  - Requires a complete understanding of how a device is going to be used and its context of use
  - Also includes concerns such as how it is maintained and what happens when it is taken out of service
  - Addresses applicable regulatory requirements
- Phase 3
  - Document outcomes of Phase 1 and Phase 2
  - (In practice, phase 2 and phase 3 often overlap)

Elaborating a specification

- Limit the search space by preparing a detailed list of all the requirements and constraints
- The following are important when preparing a specification:
  - Adopting a clear structure (such as a checklist)
  - Quantifying whenever possible
  - Identifying demands and wishes
  - Indicating sources of statements
  - Reviewing and updating regularly, and recording changes
- Where possible use quantified statements
- **Every requirement must be testable and anything relevant to a design can be made testable!**

Elaborating a specification

- To aid selection and evaluation of possible solution concepts it is useful to identify each requirements statement as being either:
  - A demand (D): **a requirement which must be fulfilled**
  - A wish (W): **a requirement which is desirable, but not essential**

- It is useful to indicate the weighting (*Wt*) of wishes as high (H), medium (M) or low (L) importance.

may use a numeric scale

- The demands in the specification provide the criteria for a preliminary selection, and the wishes provide the criteria for evaluation
- The source of a requirement should be recorded and any subsequent changes logged

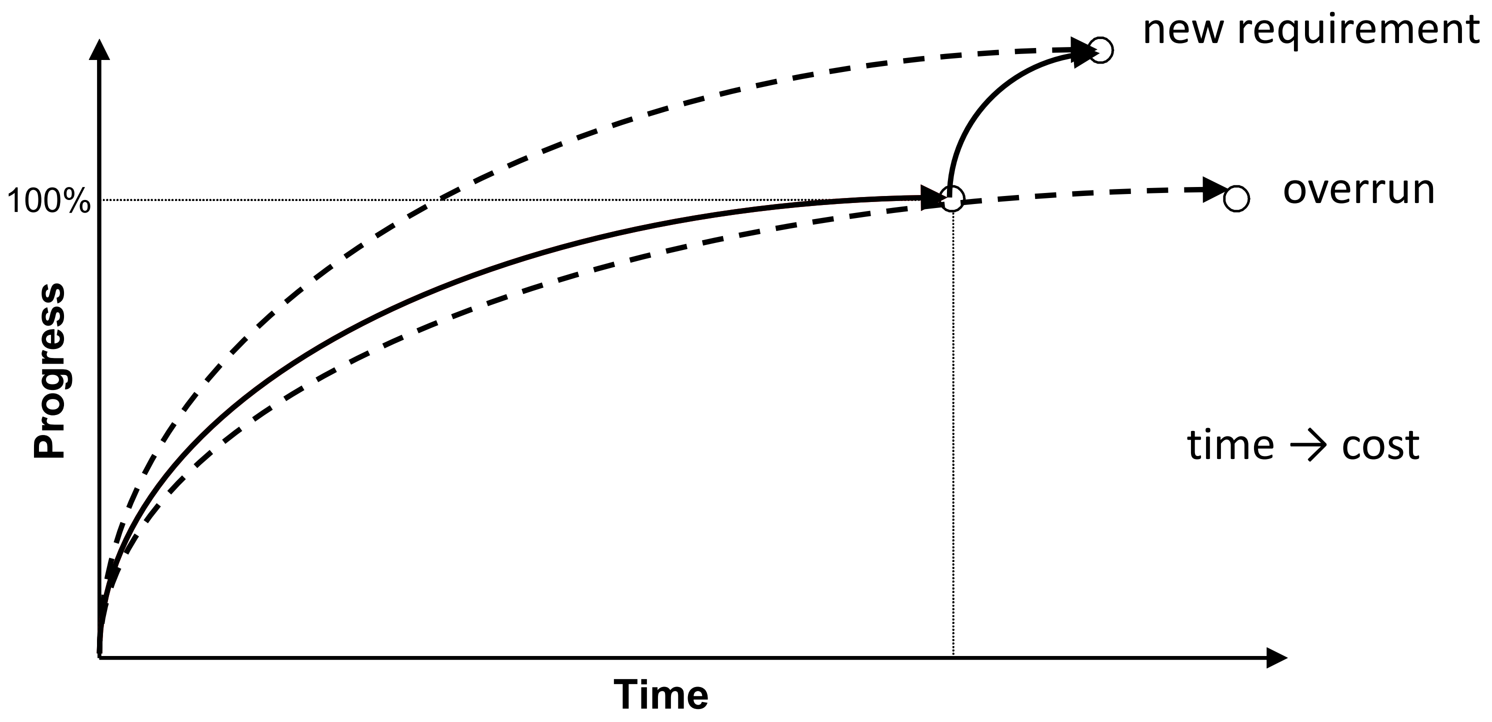
Desired Characteristics of requirements

Requirements should be:

- **Solution independent**: requirements should not specify a solution to the problem; they should specify what needs to be done, but not how it will be done
- **Complete** - the requirements must include all areas of concern, including all phases of the product life cycle
- **Clear** - requirements should not leave anyone guessing what is required
- **Concise** - unnecessary requirements should be omitted; the wording of requirements should be concise – do not bury the requirement in unneeded text
- **Testable** - Quantitative (numerical) limits, tolerances, ranges, and intended values should be indicated when possible; testable requirements can be measured in order to determine if the design goal is met
- **Traceable** –The source of a requirement should be recorded and any subsequent changes should be logged

Elaborating a specification

A specification defines a **target** for a project team to aim for and the **criteria** by which they know that they have got there. A specification is also a “**live**” document: the target often moves and progress towards it is not always straightforward



Requirements and verification

- **Verification** is the process of ensuring requirements have been met
- As such, it is important to write requirements with a clear understanding of how these can later be robustly verified
- A very common design method to address this is to write a **verification cross-reference matrix** (VCRM) in tandem with a requirements specification
  - A VCRM is a specification for how to verify each requirement, including specific verification **environments**, procedures and success criteria
- As a consequence of verification, testers are also stakeholders that should be taken into account when elaborating the requirements specification

Requirements and conceptual design

- In the conceptual design stage, the function model of the system is elaborated on and controllable and uncontrollable parameters are identified for each relevant function
  - **Controllable parameters** are design parameters that we can influence and subject to optimisation towards design objectives
  - **Uncontrollable parameters** are variables we are aware of but cannot directly influence; analysing these allow us to perform **sensitivity analysis**
- Such analyses further help shape requirements as critical parameters, such as minimum true positive rate and maximum false alarm rate, can be decided by analysis at an early stage in the design
- More on this in the next lecture!

Understanding Users

Understanding users

- Learning about users is a simple idea... yet surprisingly difficult
- **“Successful designers are aware that people learn, think, and solve problems in different ways”** (Shneiderman and Plaisant 2010)
- Therefore, all design should begin with an understanding of the intended users: **who are the end-users?**
  - Potential factors include: age, gender, physical and cognitive abilities, education, cultural or ethnic backgrounds, training, motivation, goals, personality
- User communities can be expected to carry with them a combination of usage patterns and certain knowledge
  - Examples: doctors, nurses, sys admins
- Other variables may be urban vs. rural, economic profile, disabilities, attitudes toward technology, reading skills, users’ skills with interfaces, knowledge of domain-specific information
- “...every step towards understanding the users and recognizing them as individuals with outlooks different from the designer’s own is likely to be a step closer to a successful design” (Shneiderman and Plaisant 2010)
- Various methodologies can be used to understand users (field studies, diary studies, questionnaires, etc.)

Understanding users

- Before it is possible to devise an initial requirements specification and create an initial functional design we must understand the users' needs and wants
- The primary risk in this process is **misunderstanding or misconstruing** users' needs and wants
- We also have to consider a **cost-benefit tradeoff** as the more refined the methods for understanding users' needs and wants are, the more they cost in terms of time, money, resources and opportunity costs
- A **major risk is bias,** **in particular undetected bias**: understanding the risk of bias is critical

Understanding users: a sketch of the overall process

- Identify the target audience
- Identify design objectives
- Sample users from the target audience
- Apply appropriate user research methods
- Draw conclusions
- Plan follow-up activities
  - Further user research; confirm findings, etc.

User-centred design: recruiting

- The importance of recruiting the right participants
  - Understand your **target audience**
  - Under the experience of people who are actually going to want to use, understand, and buy your product
  - Anyone else will be of marginal use and can even result in deceptive results
- Finding, inviting and scheduling the right people for your research is called **recruiting**
- The three steps of recruiting:
  1. Determining the target audience
  2. Finding representative members of that audience
  3. Convincing them to participate in your research (enticement and engagement without biasing the sampling frame)

User-centred design: interviewing

- Most user experience research is using **interviewing** as a primary research technique
- Observations are critical but to really understand user experience it is often vital to ask the user about it in an interview
- User research interviewing is not the same as interviews carried out by investigative journalists or prospective employers
- The interviewing process is more formal and standardised and it is a nondirected interview that tries to minimise the perspective of the person asking the questions

User-centred design: focus groups

- Structured, moderated group discussions
- Elicits the target audience's:
  - Conscious preferences
  - Recalled experiences
  - Stated priorities
- Information that can be gained:
  - What features people value the most and why they valu them
  - What people like best about competitor's products or services
  - Where competitor's products and services fail
  - Previously unknown competitors or applications for a product or service
- Focus groups require:
  - Good moderator
  - Careful analysis
  - Appropriate contextualisation
- Focus groups are used:
  - Early in the development cycle
  - To generate generate ideas, prioritise features and understanding the needs of the target audience

http://commons.wikimedia.org/wiki/File:Meeting.jpg

User-centred design: object-based techniques

- Some information that people know and feel about the world is difficult for people to express in words
- **Object-based techniques** supplement information that can be gained by interviewing and observing people by adding objects that participants can use as props to think with and through
- Object-based techniques are not used to answer direct research questions, rather the techniques are used to generate questions in directions the researcher has not anticipated

User-centred design: field visits

- Field visits provide information about the environment people live in that you could not otherwise get
- Meet people where they are most comfortable (habitual places and activities)
- Understand both **how** and **why** people do what they do
- Field visits clarify and focus initial project ideas by providing concrete insights into the situation, what the situation entails, and how people cope with it
- Field visits are often carried out at the very beginning of a development cycle

User-centred design: diaries

- Participants report their activities over time
- Diarists track mistakes they make, what they learn, how often they use a product, and anything else of interest to the project
- Diarists can also provide general insights into the working life of participants
- Diaries provide an unobtrusive method for gaining insight into people's lives without intrusive direct observation
- Diaries allow observation of infrequent or brief events
- Diaries reduce the time between an event and documentation (avoids asking participants to remember events of interest)

Cultural probes

- Traditional diary studies are prescriptive and ask participants to record activities accurate, as they happen
- They do not encourage imaginative personal reflection
- Cultural probes elicit such responses via structured, playful exercises
- Stimulate creative discussions between designers and potential users
- Provokes imagination and generates empathy

User-centred design: experience sampling method

- A self-report technique that enables researchers to study how people use technology in naturalistic settings
- Participants fill out several brief questionnaires every day by responding to electronic alerts
- Does not require participants to recall anything
- Researchers are decoupled which reduces bias
- Can use statistical methods to analyse data
- For a complete description of the method, see:
  - Consolvo, S. and Walker, M. 2003. Using the experience sampling method to evaluate ubicomp applications. *IEEE Pervasive Computing* **2**(2): 24-31

User-centred design: surveys

- A **survey** is a set of questions that allows a large group of people to describe themselves, their interests, and their preferences in a structured way
- A survey is usually quantitative
- Surveys reveal characteristics and user tendencies in the user population at large

Analysing qualitative data

Research questions

- Before any qualitative data analysis and data collection activity it is critical to already have an idea about the **research questions** that should be answered
  - For example, “Who visits our website and what do they value?”
- Research questions guide data collection activities and the analysis
- In particular, at the data analysis stage, initial codes are often created from research questions

A process for qualitative data analysis

1. Capture and discuss initial insights
   - Recording insights at or closely after data collection events
2. Data preparation
3. Identify patterns and themes
   - Coding
   - Affinity clustering
4. Relate groups to models (frameworks) and create stories
   - Structure data into a model that can be applied to product or service design
   - Back up all key findings and design recommendations by brief compelling stories from the data

Capturing insights while collecting data

- It is very expensive and time-consuming to transcribe audio and video recordings
  - This cost can often be avoided by diligently taking notes while recording material
- Often there are insights gained while collecting data that will be forgotten unless they are recorded immediately
  - However, it is **very** important to separate out **analysis** (your own theories about what was said) from **data** (what was actually said)
  - The **debrief** period after an interview or focus group meeting can be used to reflect about the data and capture insights close to the data collection event
  - In addition, or alternatively, researchers can discuss the data collection together immediately after the data collection event

Data preparation

- Label photos, videos, drawing, etc.
- Transcribe audio and video
- Break up large chunks of data into smaller more manageable pieces
- Add descriptions to each piece of data
- All data should be in a standardised format that can be easily be shared with team members on for instance post-its

Coding

- **Codes** are labels or tags that denote concepts
- Typically one starts out with preconceived codes that come out of the original research questions
- Each piece of data can have multiple codes
- A code can describe multiple pieces of data
- Each individual piece of data should have at least one code
- Assigning codes (and conceiving new codes) is an acquired skill that evolves as you work more and more with the data material
- During (and after) coding, **patterns** in the data will emerge
- Assigning codes require skill and careful reflection of the material **in context**

Coding for characterising people

- Values
- Mental models
- Goals
- Behaviours
- Roles
- Skill levels
- Preferred or alternative tools
- Pain points
- Demographics

Coding for the decomposition of processes in an activity

- Resources
- Mistakes/corrections
- Decision points
- Outcomes
- Frequency
- Importance
- Risks
- Purpose
- Cues
- Options

Affinity clustering

- A bottom-up approach to coding

1. Group together pieces of data that appear to be belong to each other
2. Once the groupings are stable, work out the logic as to why the groupings exist and how they relate to each other
3. Thereafter assign codes

- Often done using post-its as a group analysis activity

Other sources of codes

- Concentrate on people’s reasons
- Note people’s terminology
- Watch out for contradictions
- Watch for situations where people change their mind
- Look for stories about success or failure
- Treat words like “always” and “never” as red flags
- Don’t ignore personal judgments

Models (frameworks)

- Once there are groups think about their structure
- Should large groups be subdivided
- Is there a **model** that describes all the data or a subset of the data?
- If a piece of data no longer fits within a group, reassign it to another group
- If a group is no longer viable, break it up and re-assign the data to other groups
- Continue until you are confident your clusters show identifiable patterns

Example models (frameworks)

- Taxonomies
  - Tree hierarchies
- Maps
  - Spatial relationships
- Timelines
  - For instance, plotting the occurrence of a particular code in a diary study over time
- Spectrums
  - Visualising a single dimension
- 2-by-2 matrices
  - Visualising two dimensions

Stories

- Stories connect models to the original data and the people who generated that data
- A successful qualitative analysis consists of both models that allow people to draw out actionable recommendations and stories that give your models credibility based on evidence from the data
- Stories provide details about:
  - People's assumptions
  - The sequences in which people do things
  - People's problems and approaches to solving them
  - People's opinions
- Stories prompt **informed empathy and invention**
- It is important to be able to link back compelling stories collected in the data to codes and your models

Sample bias, drawing conclusions and follow-up activities

Sampling bias

- Sampling bias: members of your sampling frame are not members of your population
- A **severe** risk of error that can result in downright dangerously deceptive research conclusions
- People will have to be excluded but it is important to realise who are excluded and why

Sampling bias types

- Non-responder bias
- Timing bias
- Duration bias
- Invitation bias
- Self-selection
- Presentation bias
- Expectation bias

Drawing conclusions

- Common problems to avoid:
  - Confusing correlation and causation
  - Not differentiating between subpopulations
  - Confusing belief with truth
- Issues to take into account:
  - People want everything
  - People exaggerate
  - People will choose an answer even if they don’t feel strongly about it
  - People try to outguess the survey
  - People lie

Follow-up activities

- Follow-up qualitative research
  - Nondirected interviews
  - Focus groups
  - Field trips
- Tracking surveys
  - Running a similar survey at regular intervals
- Refined surveys
  - Narrowing in on particular issues identified in the first survey
- Pre/post surveys
  - For example, before and after a major redesign

Summary

- Design engineering is a useful *complementary* methodology when the complexity of the system is high
- To do any sensible design work we must evolve a **solution-neutral problem statement**
- To do this we must understand the **target audience** and users' needs and wants
- **Requirements** constrain the design space and implicitly specify the tests that the system must succeed to ultimately pass **verification**
- Requirements evolve during the design process and analysis of functions and function carriers in the conceptual design stage is likely to affect the final requirements specification