Identifying needs and establishing requirements

Overview
- The importance of requirements
- Different types of requirements
- Data gathering for requirements
- Task descriptions: Scenarios
  Use Cases
  Essential use cases
- Task analysis: HTA

What, how and why?
- **What**
  Two aims:
  1. Understand as much as possible about users, task, context
  2. Produce a stable set of requirements
- **How**
  Data gathering activities
  Data analysis activities
  Expression as ‘requirements’
  All of this is iterative

What, how and why?
- **Why**:
  Requirements definition: the stage where failure occurs most commonly
  Getting requirements right is **crucial**

Establishing requirements
- **What do users want? What do users 'need'?**
  Requirements need clarification, refinement, completion, re-scoping
  Input: requirements document (maybe)
  Output: stable requirements
- **Why 'establish'?**
  Requirements arise from understanding users' needs
  Requirements can be justified & related to data

Different kinds of requirements
- **Functional:**
  —What the system should do
  —Historically the main focus of requirements activities
- **(Non-functional):** memory size, response time...
- **Data:**
  —What kinds of data need to be stored?
  —How will they be stored (e.g. database)?
Different kinds of requirements

Environment or context of use:

— physical: dusty? noisy? vibration? light? heat? humidity? ... (e.g. OMS insects, ATM)
— social: sharing of files, of displays, in paper, across great distances, work individually, privacy for clients
— organisational: hierarchy, IT department’s attitude and remit, user support, communications structure and infrastructure, availability of training

Kinds of requirements

What factors (environmental, user, usability) would affect the following systems?

• Self-service filling and payment system for a petrol (gas) station
• On-board ship data analysis system for geologists searching for oil
• Fashion clothes website

Personas

• Capture user characteristics
• Not real people, but synthesised from real user characteristics
• Should not be idealised
• Bring them to life with a name, characteristics, goals, personal background
• Develop multiple personas

Data gathering for requirements

Interviews:

— Props, e.g. sample scenarios of use, prototypes, can be used in interviews
— Good for exploring issues
— But are time consuming and may be infeasible to visit everyone

Focus groups:

— Group interviews
— Good at gaining a consensus view and/or highlighting areas of conflict
— But can be dominated by individuals
Data gathering for requirements

Questionnaires:
— Often used in conjunction with other techniques
— Can give quantitative or qualitative data
— Good for answering specific questions from a large, dispersed group of people

Researching similar products:
— Good for prompting requirements

Direct observation:
— Gain insights into stakeholders’ tasks
— Good for understanding the nature and context of the tasks
— But, it requires time and commitment from a member of the design team, and it can result in a huge amount of data

Indirect observation:
— Not often used in requirements activity
— Good for logging current tasks

Data gathering for requirements

Studying documentation:
— Procedures and rules are often written down in manuals
— Good source of data about the steps involved in an activity, and any regulations governing a task
— Not to be used in isolation
— Good for understanding legislation, and getting background information
— No stakeholder time, which is a limiting factor on the other techniques

Contextual Inquiry
• An approach to ethnographic study where user is expert, designer is apprentice
• A form of interview, but
  — at users’ workplace (workstation)
  — 2 to 3 hours long
• Four main principles:
  — Context: see workplace & what happens
  — Partnership: user and developer collaborate
  — Interpretation: observations interpreted by user and developer together
  — Focus: project focus to understand what to look for

Some examples
Future Technology Workshops: Interacting with images
The Living Box: Ethnographic interviews, focus groups with props, and questionnaires

Problems with data gathering (1)
• Identifying and involving stakeholders: users, managers, developers, customer reps?, union reps?, shareholders?
• Involving stakeholders: workshops, interviews, workplace studies, co-opt stakeholders onto the development team
• ‘Real’ users, not managers: traditionally a problem in software engineering, but better now
Problems with data gathering (2)

- Requirements management: version control, ownership
- Communication between parties:
  — within development team
  — with customer/user
  — between users... different parts of an organisation use different terminology
- Domain knowledge distributed and implicit:
  — difficult to dig up and understand
  — knowledge articulation: how do you walk?
- Availability of key people

Problems with data gathering (3)

- Political problems within the organisation
- Dominance of certain stakeholders
- Economic and business environment changes
- Balancing functional and usability demands

Some basic guidelines

- Focus on identifying the stakeholders’ needs
- Involve all the stakeholder groups
- Involve more than one representative from each stakeholder group
- Use a combination of data gathering techniques

Some basic guidelines

- Support the process with props such as prototypes and task descriptions
- Run a pilot session
- You will need to compromise on the data you collect and the analysis to be done, but before you can make sensible compromises, you need to know what you’d really like
- Consider carefully how to record the data

Data interpretation and analysis

- Start soon after data gathering session
- Initial interpretation before deeper analysis
- Different approaches emphasize different elements e.g. class diagrams for object-oriented systems, entity-relationship diagrams for data intensive systems

Task descriptions

- Scenarios
  — an informal narrative story, simple, ‘natural’, personal, not generalisable
- Use cases
  — assume interaction with a system
  — assume detailed understanding of the interaction
- Essential use cases
  — abstract away from the details
  — does not have the same assumptions as use cases
Scenario for holiday planner

"The Thomson family enjoy outdoor activity holidays and want to try their hand at sailing this year. There are four members of the family: Sky who is 10 years old, Eamonn who is 15 years old, Claire who is 35, and Will who is 40. While out on a shopping trip they call by at the travel agents in their local town to start exploring the possibilities ... The travel organizer is located in a quiet corner of the agents’ office, where there are comfortable seats and play things for young children. They all gather around the organizer and enter their initial set of requirements—a sailing holiday for four novices. The stand-alone console is designed so that all members of the family can interact easily and comfortably with it. The system’s initial suggestion is that they should consider a flotilla holiday, where several novice crews go sailing together and provide mutual support for first-time sailors..."

Use case for holiday planner

1. The system displays options for investigating visa and vaccination requirements.
2. The user chooses the option to find out about visa requirements.
3. The system prompts user for the name of the destination country.
4. The user enters the country’s name.
5. The system checks that the country is valid.
6. The system prompts the user for her nationality.
7. The user enters her nationality.
8. The system checks the visa requirements of the entered country for a passport holder of her nationality.
9. The system displays the visa requirements.
10. The system displays the option to print out the visa requirements.
11. The user chooses to print the requirements.

Alternative courses for holiday planner

Some alternative courses:

6. If the country name is invalid:
   6.1 The system displays an error message.
   6.2 The system returns to step 3.

8. If the nationality is invalid:
   8.1 The system displays an error message.
   8.2 The system returns to step 6.

9. If no information about visa requirements is found:
   9.1 The system displays a suitable message.
   9.2 The system returns to step 1.

Example use case diagram for holiday planner

Example essential use case case for holiday planner

```
retrieveVisa

<table>
<thead>
<tr>
<th>USER INTENTION</th>
<th>SYSTEM RESPONSIBILITY</th>
</tr>
</thead>
<tbody>
<tr>
<td>find visa requirements</td>
<td>request destination and nationality</td>
</tr>
<tr>
<td>supply required information</td>
<td>obtain appropriate visa info</td>
</tr>
<tr>
<td>obtain copy of visa info</td>
<td>offer info in different formats</td>
</tr>
<tr>
<td>choose suitable format</td>
<td>provide info in chosen format</td>
</tr>
</tbody>
</table>
```

Task analysis

- Task descriptions are often used to envision new systems or devices
- Task analysis is used mainly to investigate an existing situation
- It is important not to focus on superficial activities
- What are people trying to achieve?
- Why are they trying to achieve it?
- How are they going about it?
- Many techniques, the most popular is Hierarchical Task Analysis (HTA)
### Hierarchical Task Analysis

- Involves breaking a task down into subtasks, then sub-sub-tasks and so on. These 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.
- Start with a user goal which is examined and the main tasks for achieving it are identified.
- Tasks are sub-divided into sub-tasks.

### Example Hierarchical Task Analysis

0. In order to borrow a book from the library
   1. go to the library
   2. find the required book
      - 2.1 access library catalogue
      - 2.2 access the search screen
      - 2.3 enter search criteria
      - 2.4 identify required book
   3. go to correct shelf and retrieve book
   4. take book to checkout counter

### Example Hierarchical Task Analysis (plans)

- Plan 0: do 1-3-4. If book isn’t on the shelf expected, do 2-3-4.
- Plan 2: do 2.1-2.4-2.5. If book not identified do 2.2-2.3-2.4.

### Example Hierarchical Task Analysis (graphical)

```
0. Borrow a book from the library
   1. go to the library
   2. find required book
      - 2.1 access library catalogue
      - 2.2 access the search screen
      - 2.3 enter search criteria
      - 2.4 identify required book
   3. go to correct shelf and retrieve book
   4. take book to checkout counter
```

### Summary

- Getting requirements right is crucial.
- There are different kinds of requirement, each is significant for interaction design.
- The most commonly-used techniques for data gathering are: questionnaires, interviews, focus groups, direct observation, studying documentation and researching similar products.
- Scenarios, use cases and essential use cases can be used to articulate existing and envisioned work practices.
- Task analysis techniques such as HTA help to investigate existing systems and practices.