

# Identifying needs and establishing requirements



1

## Overview

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



2

## *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

3

## *What, how and why?*

- Why:
  - Requirements definition: the stage where failure occurs most commonly
  - Getting requirements right is crucial

4

## *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

5

## *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)?

6

## *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

7

## *Different kinds of requirements*

- Users: Who are they?
  - Characteristics: ability, background, attitude to computers
  - System use: novice, expert, casual, frequent
  - Novice: step-by-step (prompted), constrained, clear information
  - Expert: flexibility, access/power
  - Frequent: short cuts
  - Casual/infrequent: clear instructions, e.g. menu paths

8

## *Different kinds of requirements*

- Usability:  
learnability, throughput, flexibility,  
attitude

Note that user requirements and usability requirements refer to different things

9

## *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

10

## *Data gathering techniques (1)*

Questionnaires:

- A series of questions designed to elicit specific information
- Questions may require different kinds of answers:  
simple YES/NO; choice of pre-supplied answers; comment
- 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

11

## *Data gathering techniques (2)*

Interviews:

- Forum for talking to people
- Structured, unstructured or semi-structured
- 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

12

## *Data gathering techniques (3)*

Workshops or focus groups:

- Group interviews
- Good at gaining a consensus view and/or highlighting areas of conflict

13

## *Data gathering techniques (4)*

Naturalistic observation:

- Spend time with stakeholders in their day-to-day tasks, observing work as it happens
- 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
- Ethnography is one form

14

## *Data gathering techniques (5)*

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

15

## *Choosing between techniques*

Data gathering techniques differ in two ways:

1. Amount of time, level of detail and risk associated with the findings
2. Knowledge the analyst requires

The choice of technique is also affected by the kind of task to be studied:

- Sequential steps or overlapping series of subtasks?
- High or low, complex or simple information?
- Task for a layman or a skilled practitioner?

16

## *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

17

## *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

18

## *Problems with data gathering (3)*

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

19

## *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

20

## *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

21

## *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

22

## *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

23

## *Scenario for shared calendar*

"The user types in all the names of the meeting participants together with some constraints such as the length of the meeting, roughly when the meeting needs to take place, and possibly where it needs to take place. The system then checks against the individuals' calendars and the central departmental calendar and presents the user with a series of dates on which everyone is free all at the same time. Then the meeting could be confirmed and written into people's calendars. Some people, though, will want to be asked before the calendar entry is made. Perhaps the system could email them automatically and ask that it be confirmed before it is written in."

24

## Use case for shared calendar

1. The user chooses the option to arrange a meeting.
2. The system prompts user for the names of attendees.
3. The user types in a list of names.
4. The system checks that the list is valid.
5. The system prompts the user for meeting constraints.
6. The user types in meeting constraints.
7. The system searches the calendars for a date that satisfies the constraints.
8. The system displays a list of potential dates.
9. The user chooses one of the dates.
10. The system writes the meeting into the calendar.
11. The system emails all the meeting participants informing them of their appointment

25

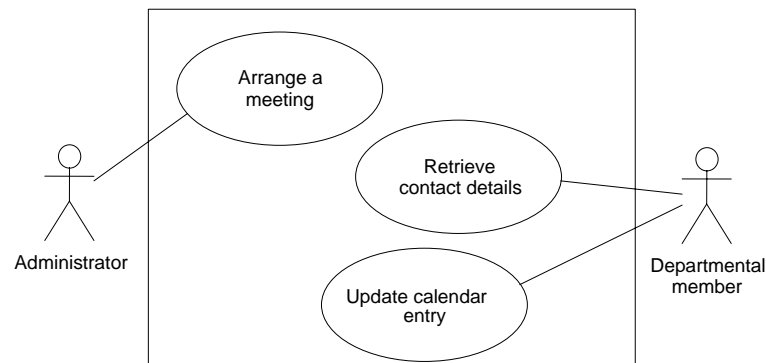
## Alternative courses for shared calendar

Some alternative courses:

5. If the list of people is invalid,
  - 5.1 The system displays an error message.
  - 5.2 The system returns to step 2.
8. If no potential dates are found,
  - 8.1 The system displays a suitable message.
  - 8.2 The system returns to step 5.

26

## Example use case diagram for shared calendar



27

## Example essential use case for shared calendar

arrangeMeeting USER INTENTION	SYSTEM RESPONSIBILITY
arrange a meeting	request meeting attendees & constraints
identify meeting attendees & constraints	search calendars for suitable dates suggest potential dates
choose preferred date	book meeting

28

## *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)

29

## *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

30

## *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
    - 2.5 note location
  3. go to correct shelf and retrieve book
  4. take book to checkout counter

31

## *Example Hierarchical Task Analysis (plans)*

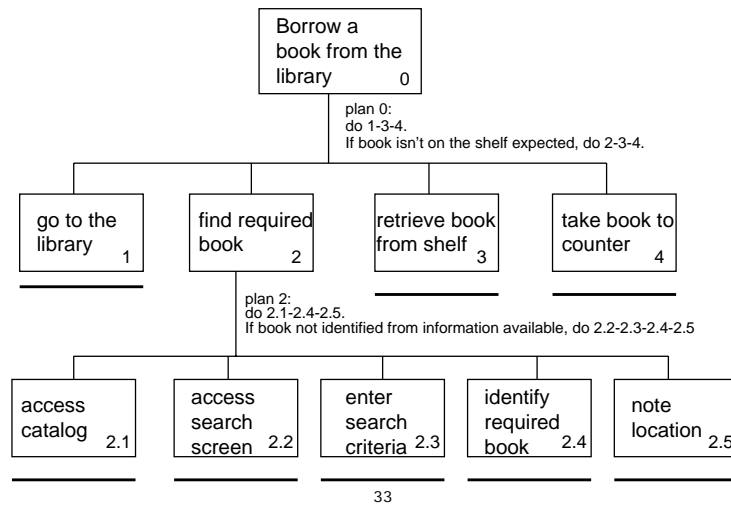
plan 0: do 1-3-4. If book isn't on the shelf expected, do 2-3-4.

plan 2: do 2.1-2.4-2.5. If book not identified do 2.2-2.3-2.4.

32



## Example Hierarchical Task Analysis (graphical)



## 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 and workshops, naturalistic observation, studying documentation
- 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