# User Center Design and Development of Interactive Environments and the WEB

Costas Mourlas

Associate Professor

Univ. of Athens

2018-2019

User-centred design process

1. Identify needs and establish requirements
2. Generate alternative solutions/designs
3. Build interactive prototypes that can be communicated and assessed
4. Evaluating design
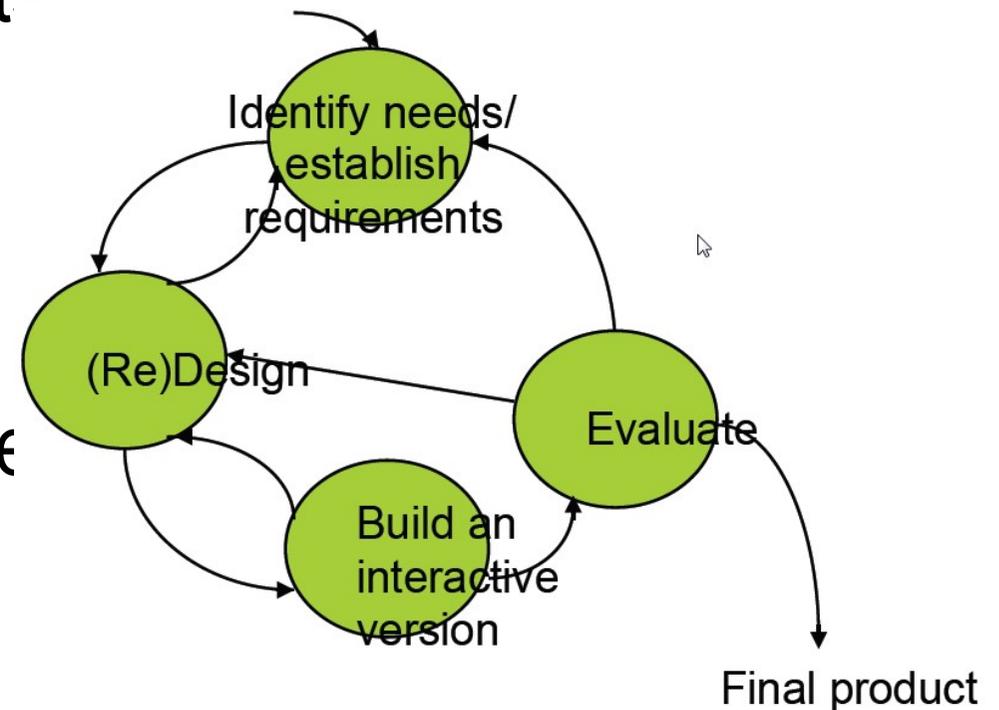
User centred design

The UCD Methodology

Step-by-Step Usability Guide

We are here

Plan → Develop a Plan → Assemble Project Team → Hold Kick-off Meeting

Write a Statement of Work

Hire a Usability Specialist

Analyze → Evaluate Your Current Site → Learn About Your Users → Conduct Task Analysis → Develop Personas → Write Scenarios → Set Measurable Goals

Design → Determine Site Requirements → Conduct a Content Inventory → Perform Card Sorting → Define IA → Write for the Web → Use Parallel Design → Develop a Prototype → Launch Site

Test and Refine → Types of Evaluations → Usability Testing → Heuristic Evaluations → Implement and Retest

Create Test Plan → Prepare and Test → Data Analyses and Report

Our Next Steps

- **Analysis: User Requirements Analysis**
  - Task Analysis → Functionality
 - Content requirements → Content
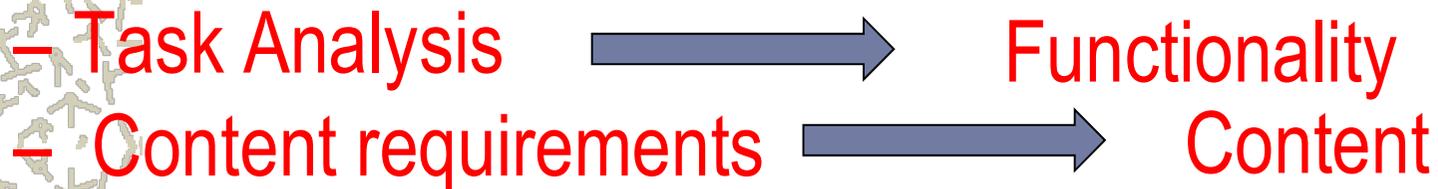
- Design: define navigation, visual design, prototypes, generate alternatives, evaluate

- Test and Refine

- Implementation

- Final Evaluation
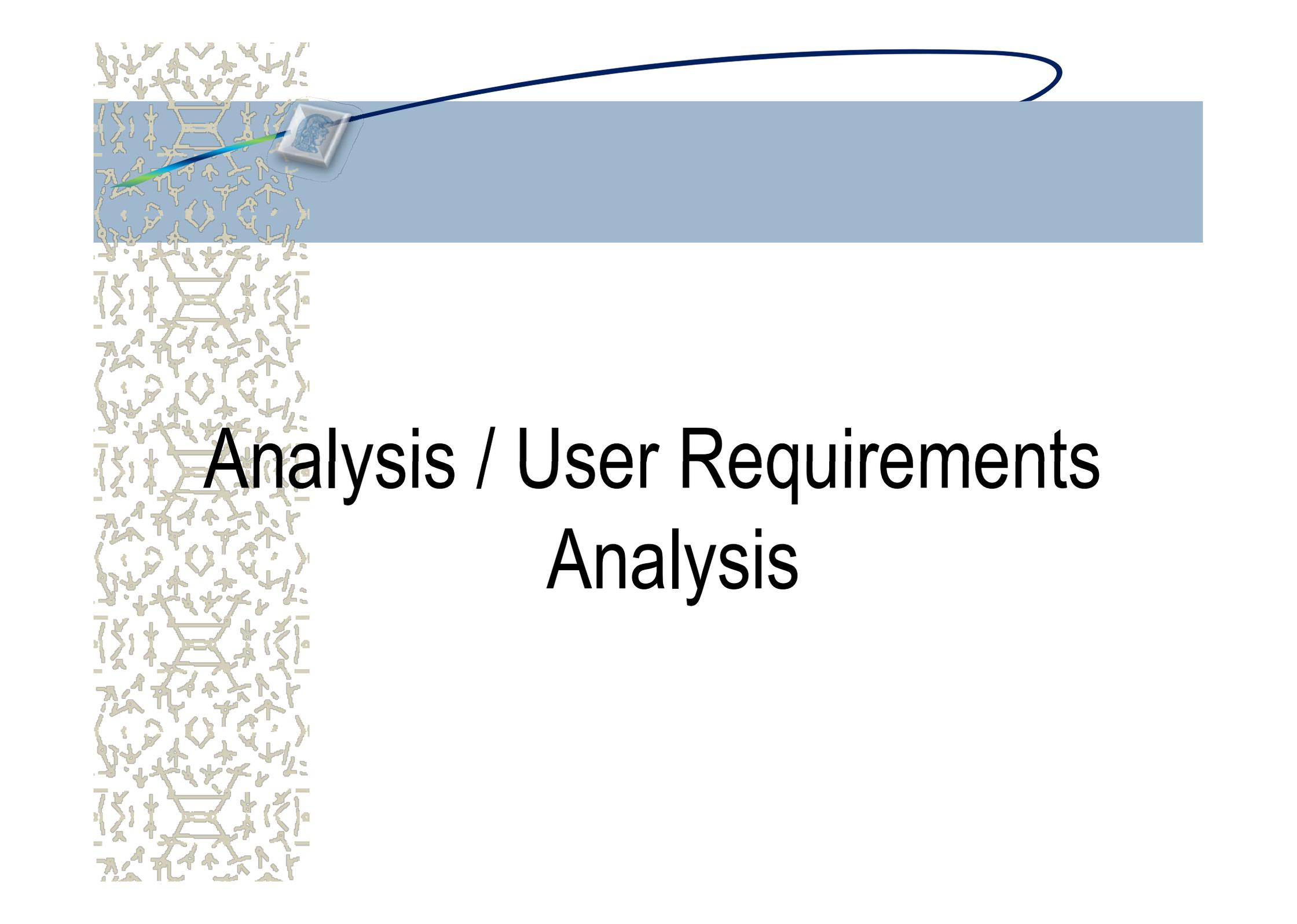
Analysis / User Requirements Analysis

Overview

- The importance of requirements

- Different types of requirements

- Data gathering for requirements

- Task descriptions: Scenarios

  Use Cases

  Essential use cases

- Task analysis: HTA

6
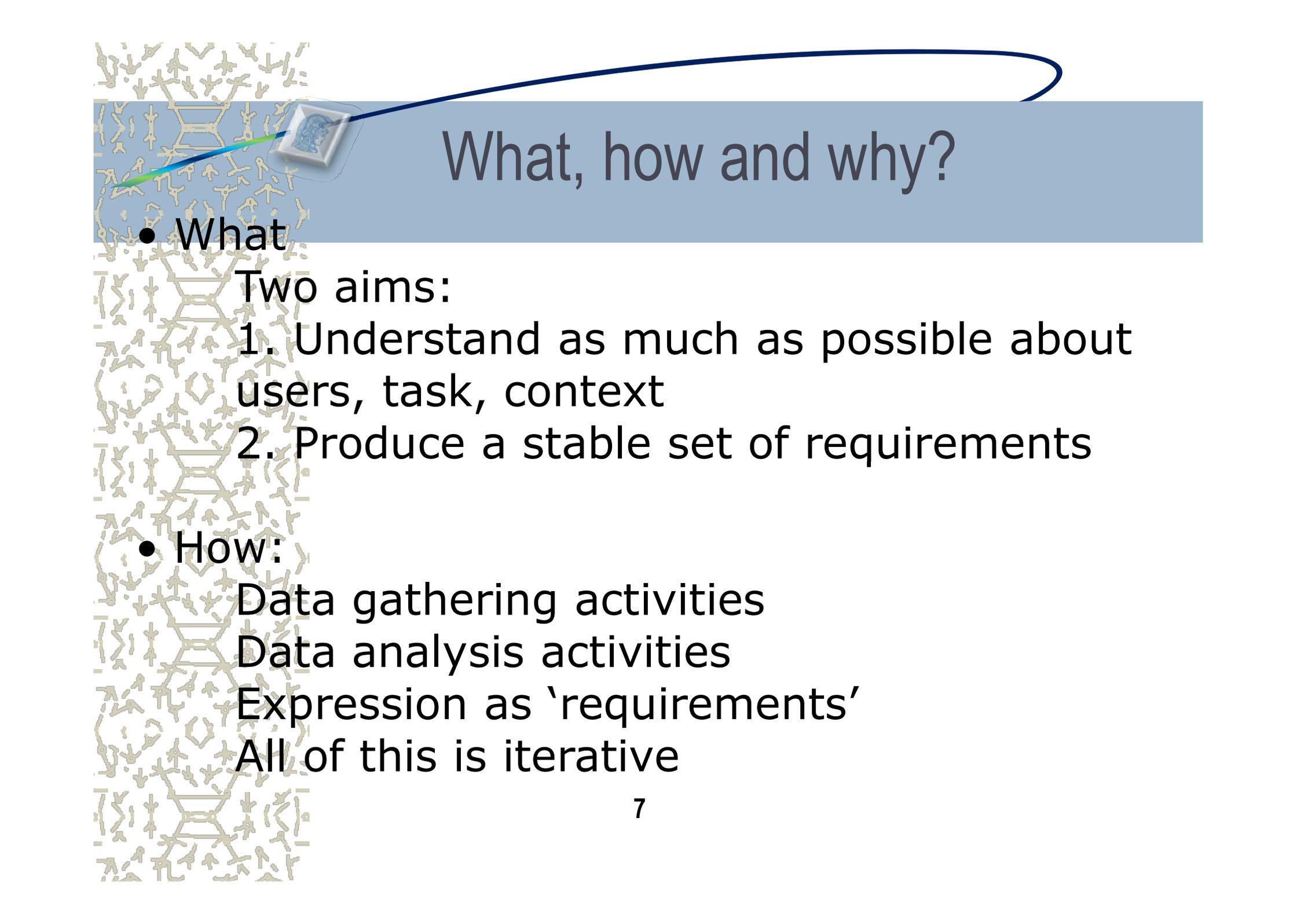
What, how and why?

- 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

• Why:

Requirements definition: the stage where failure occurs most commonly
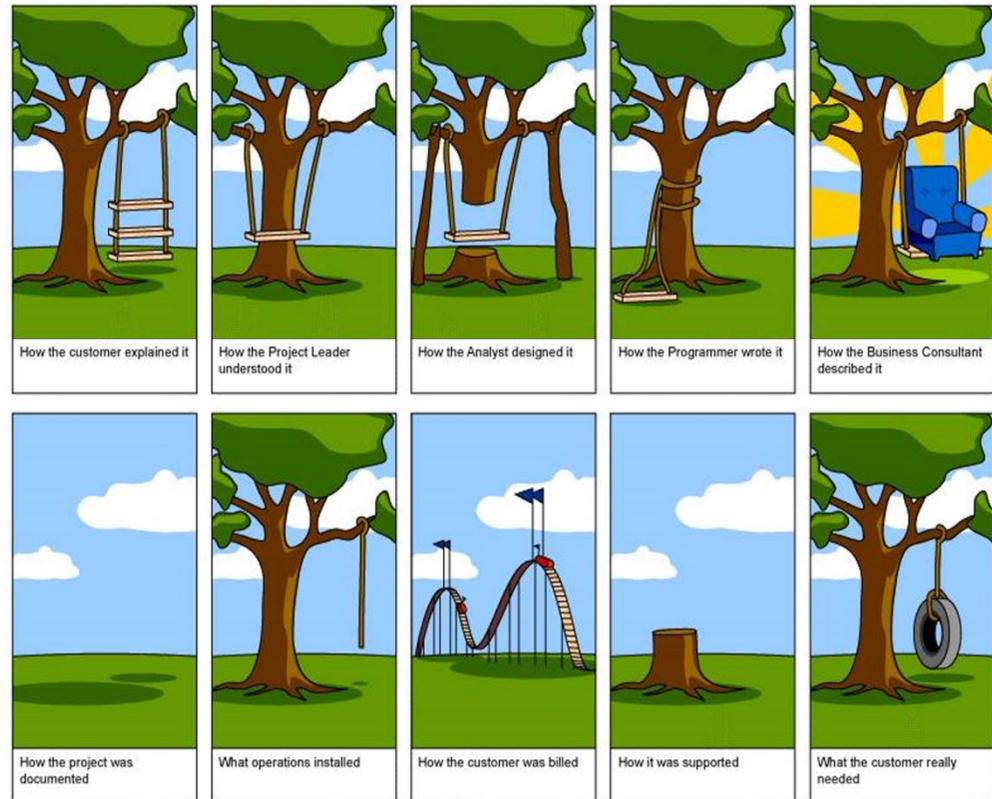
Getting requirements right is <u>crucial</u>

Establishing requirements

- What do users want? What do users 'need'?

  Requirements need clarification, refinement, completion, re-scoping

  Input: Personas + 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

- Content/Information/Data:
  - What kinds of data need to be stored?
  - How will they be stored (e.g. database)?
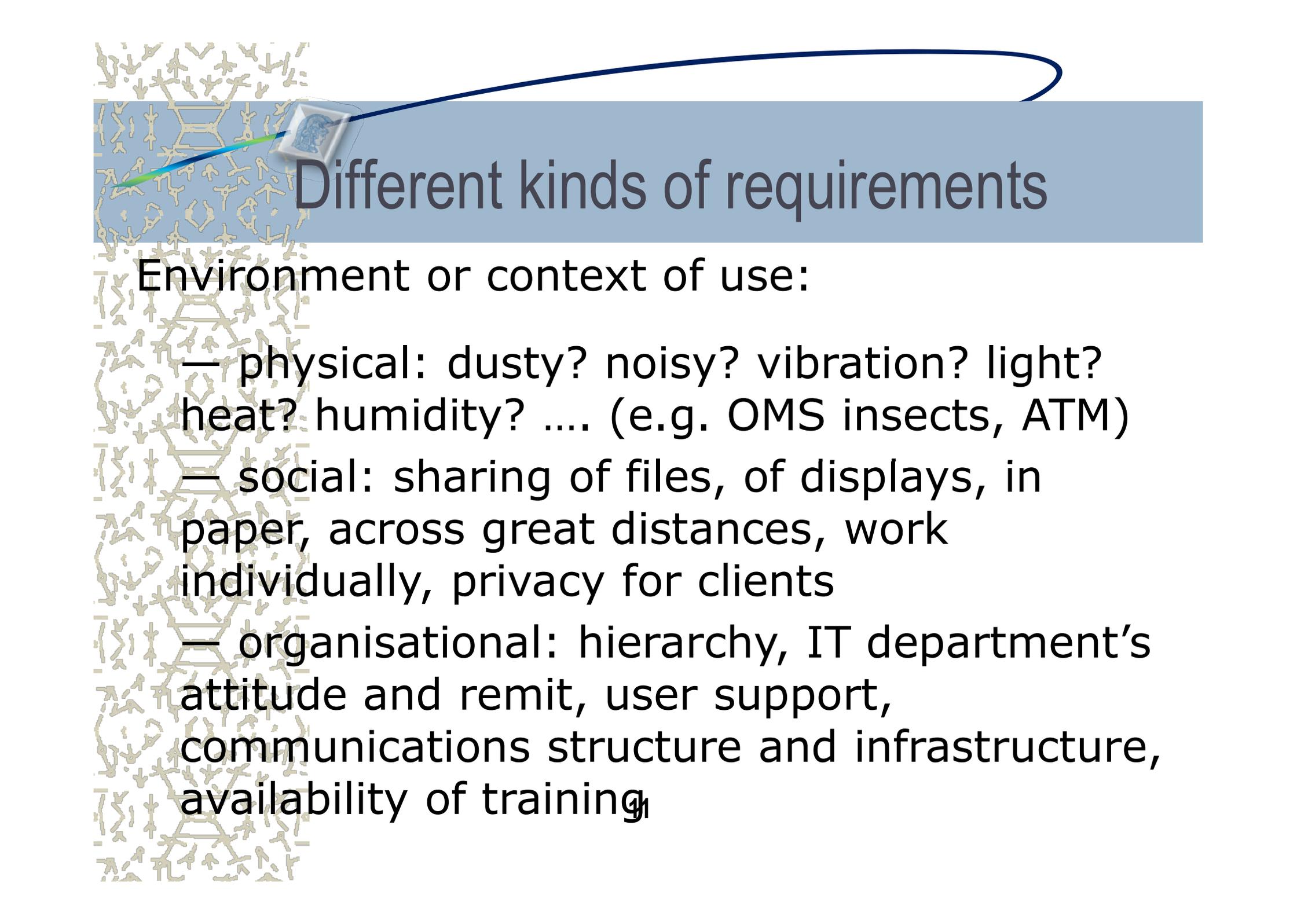
Different kinds of requirements

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
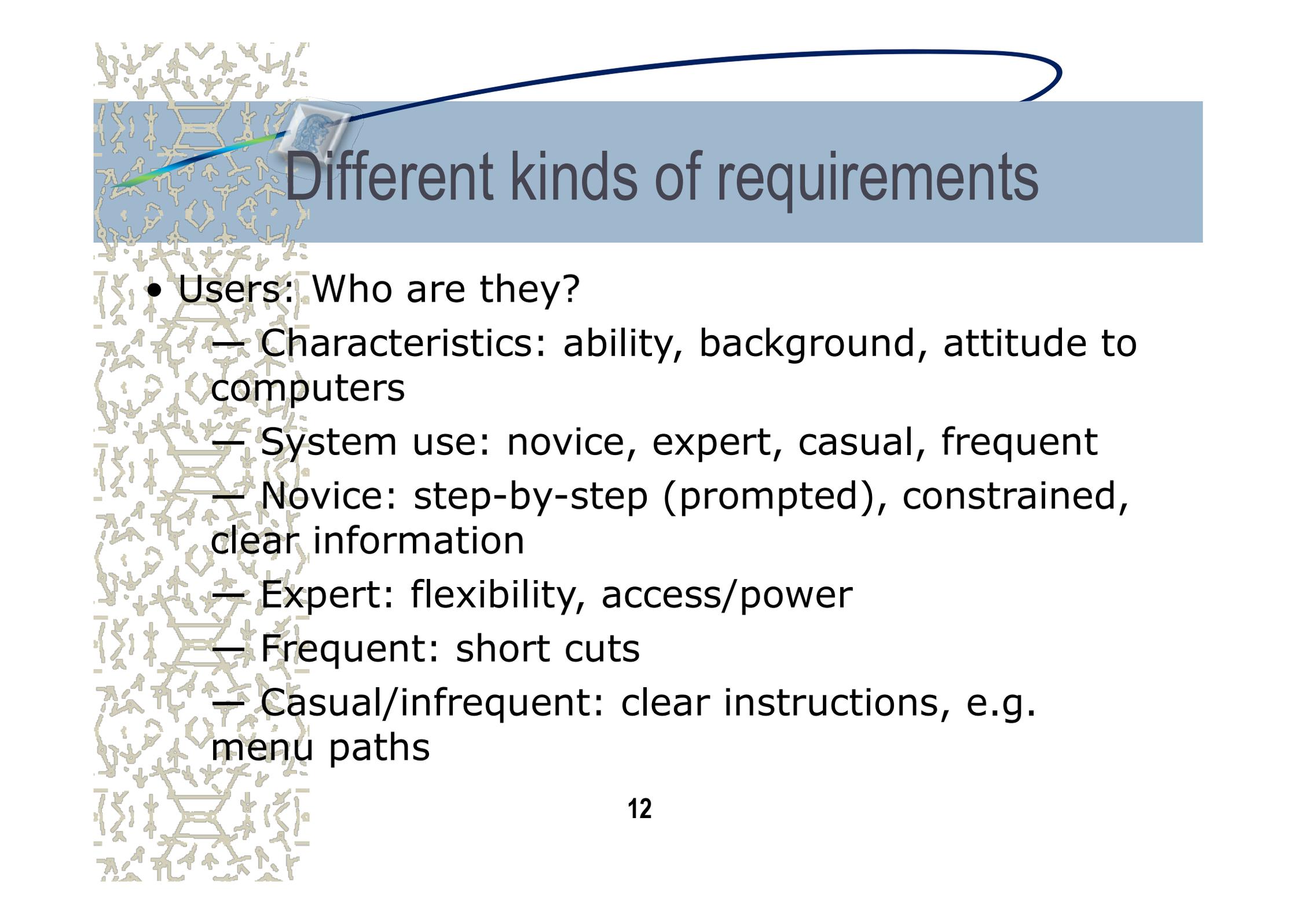
Different kinds of requirements

- 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

What are the users' capabilities?

Humans vary in many dimensions:

— size of hands may affect the size and positioning of input buttons

— motor abilities may affect the suitability of certain input and output devices

— height if designing a physical kiosk

— strength - a child's toy requires little strength to operate, but greater strength to change batteries

— disabilities (e.g. sight, hearing, dexterity)
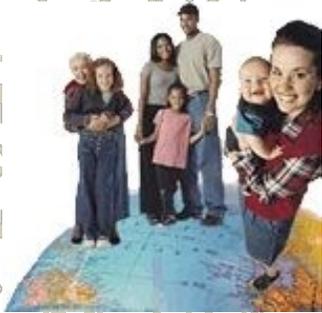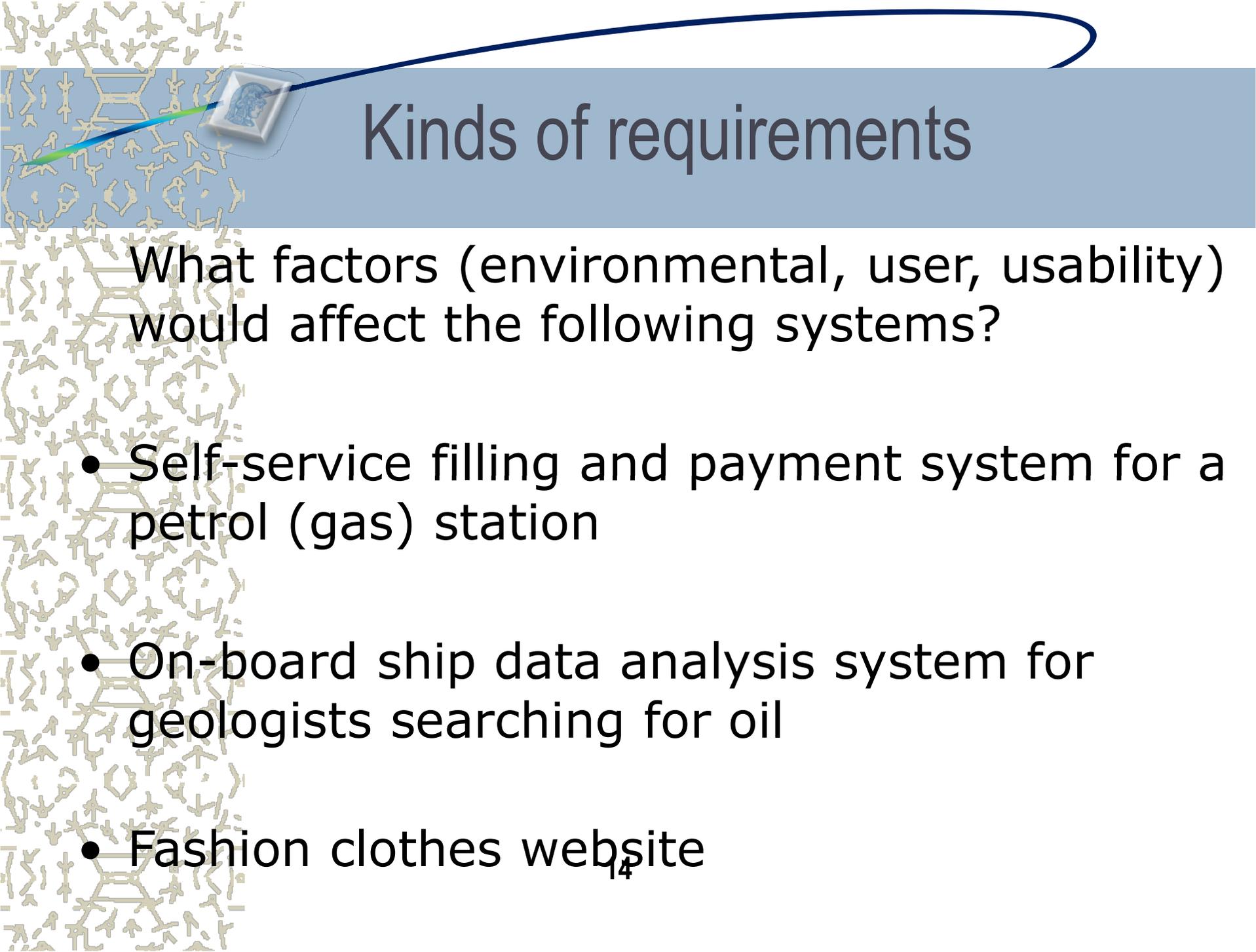
Kinds of requirements

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

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
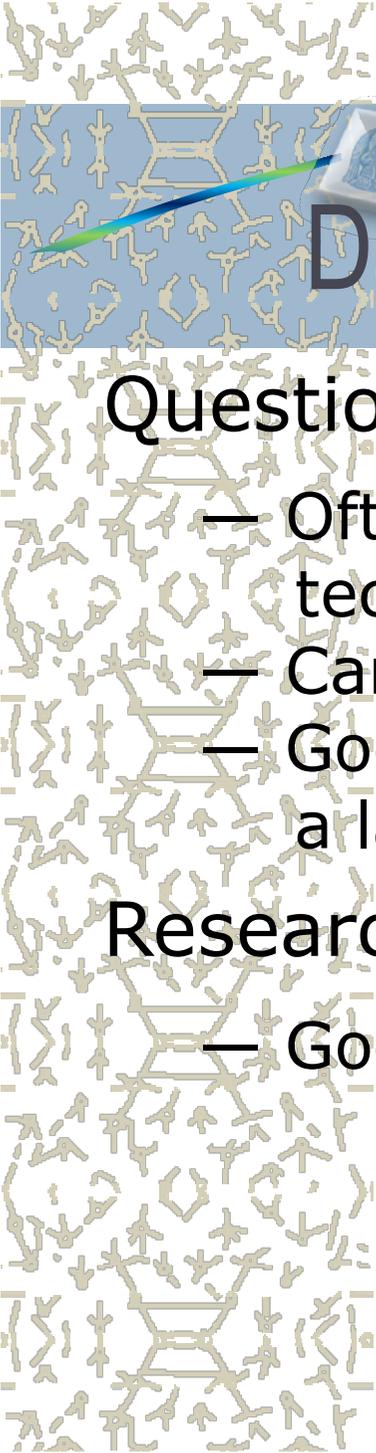
Data gathering for requirements

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
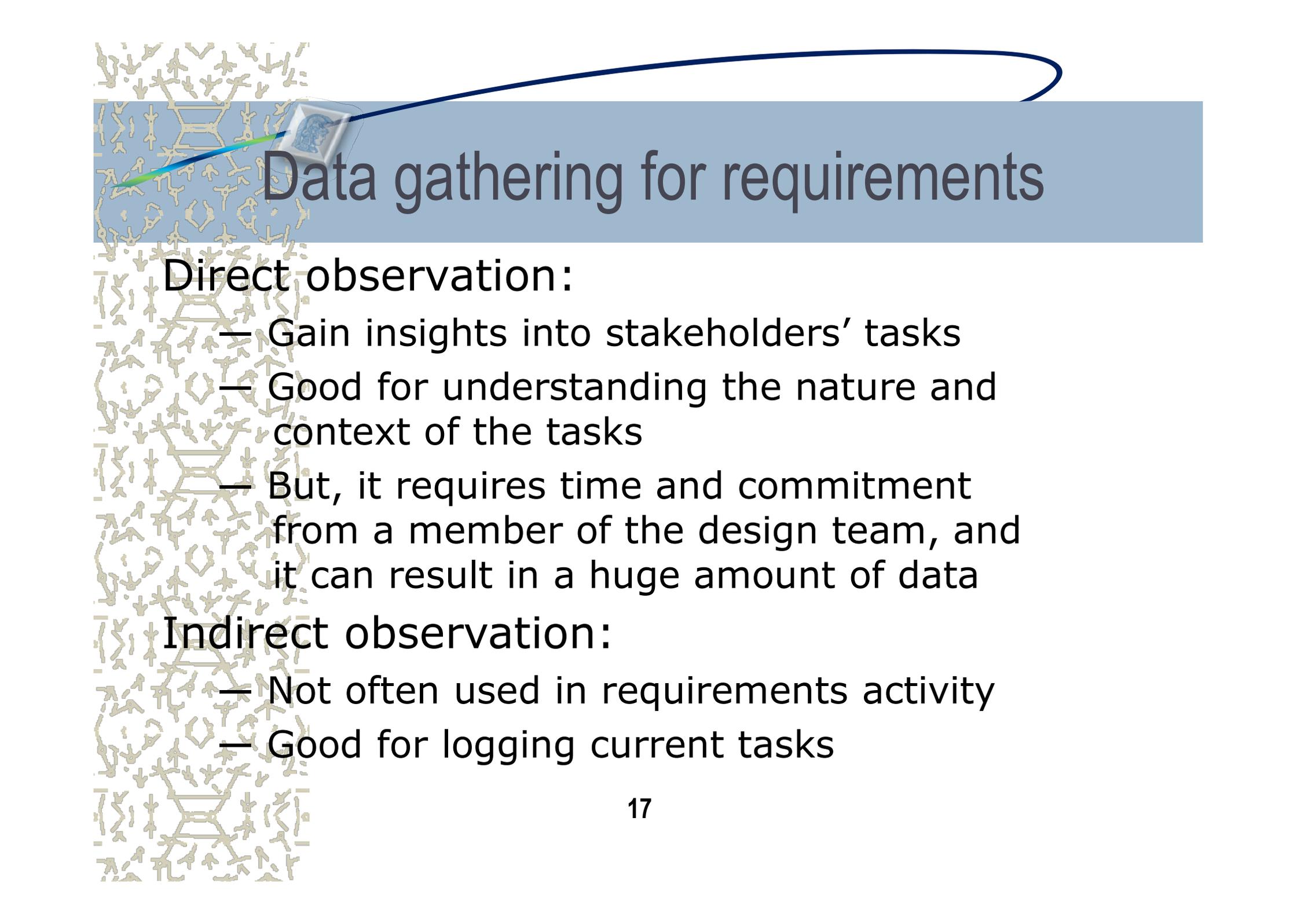
Data gathering for requirements

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

17
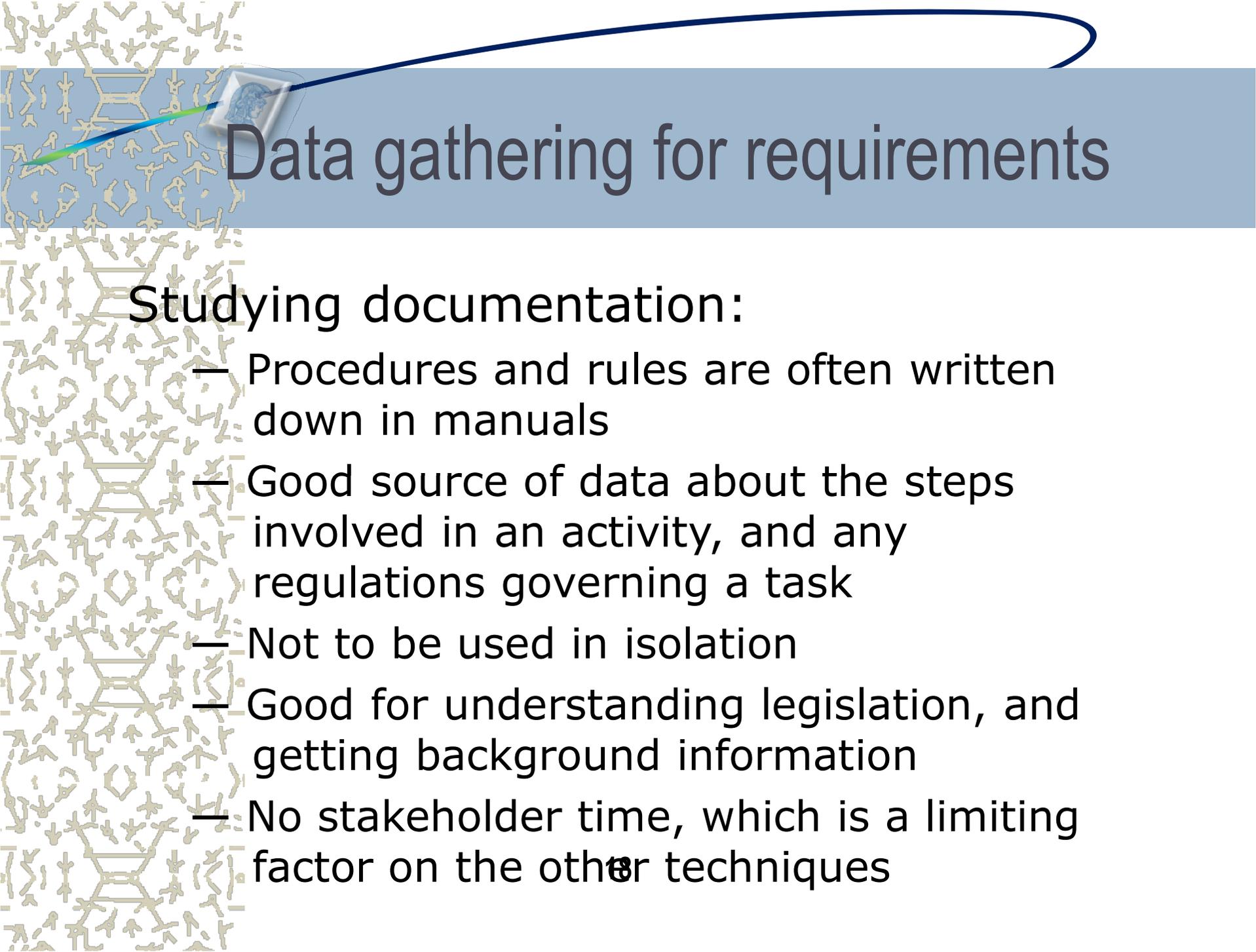
Data gathering for requirements

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

Some examples

Diary and interview

Cultural probes

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
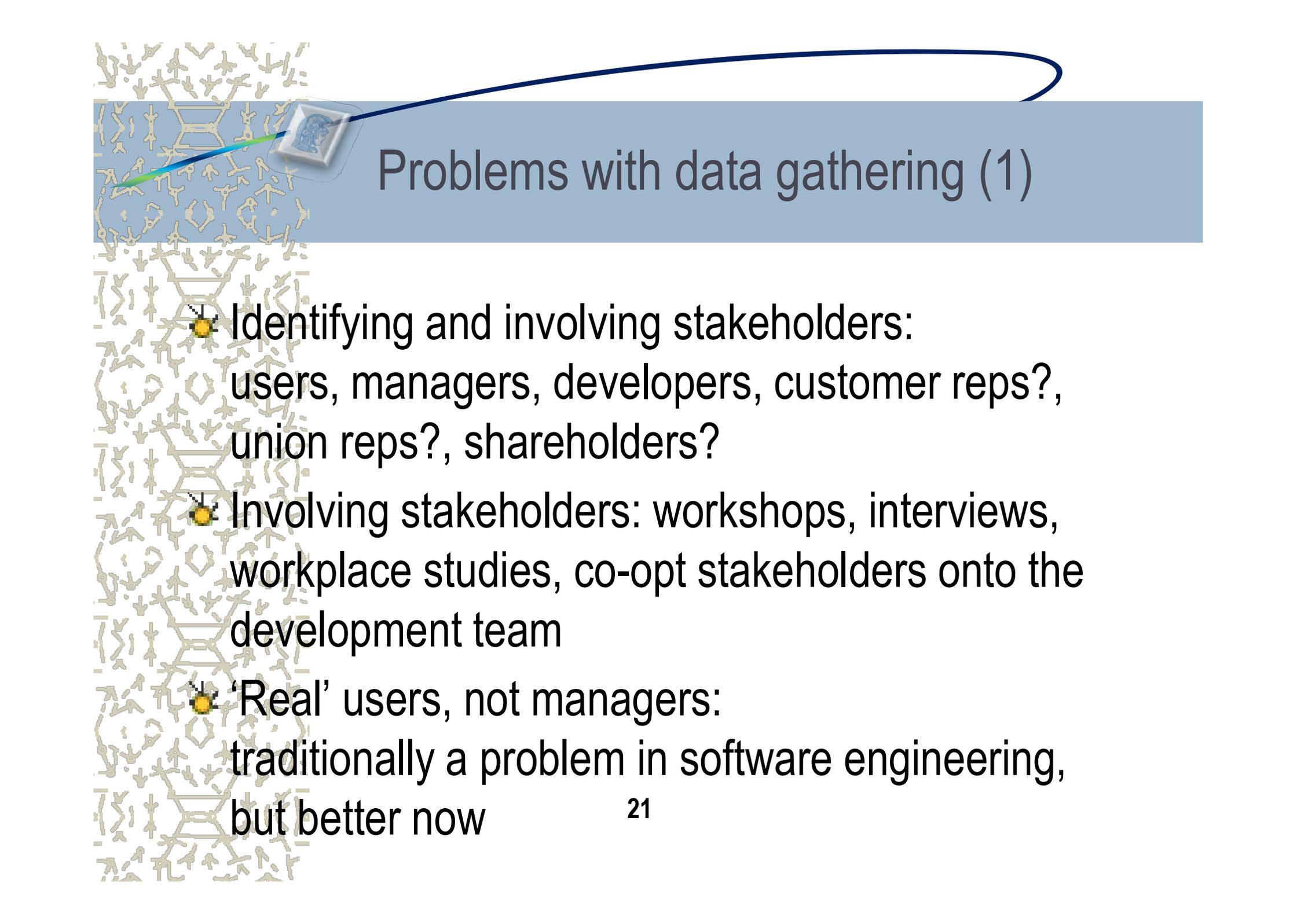
Problems with data gathering (1)

- 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

21
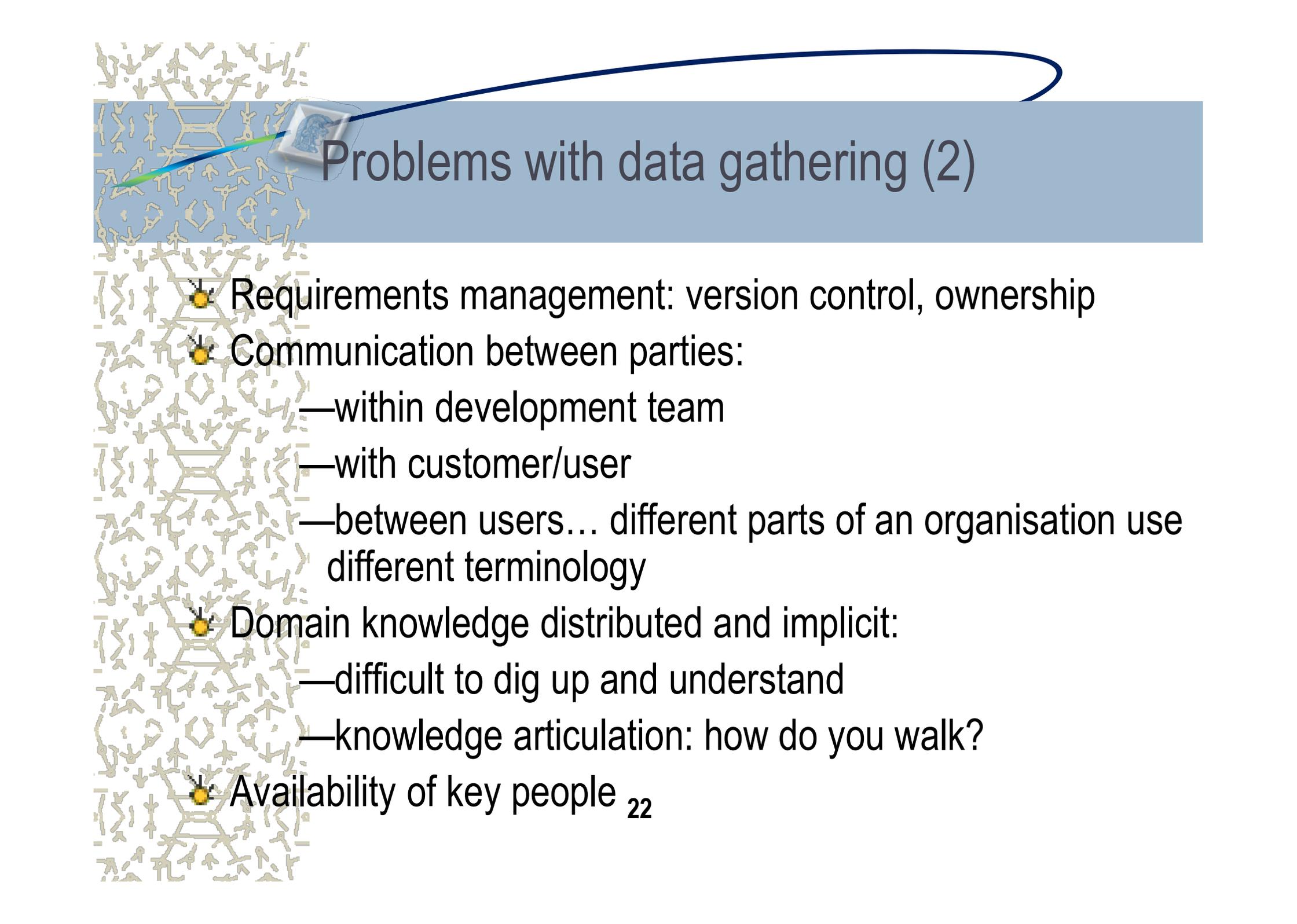
Problems with data gathering (2)

- 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 22
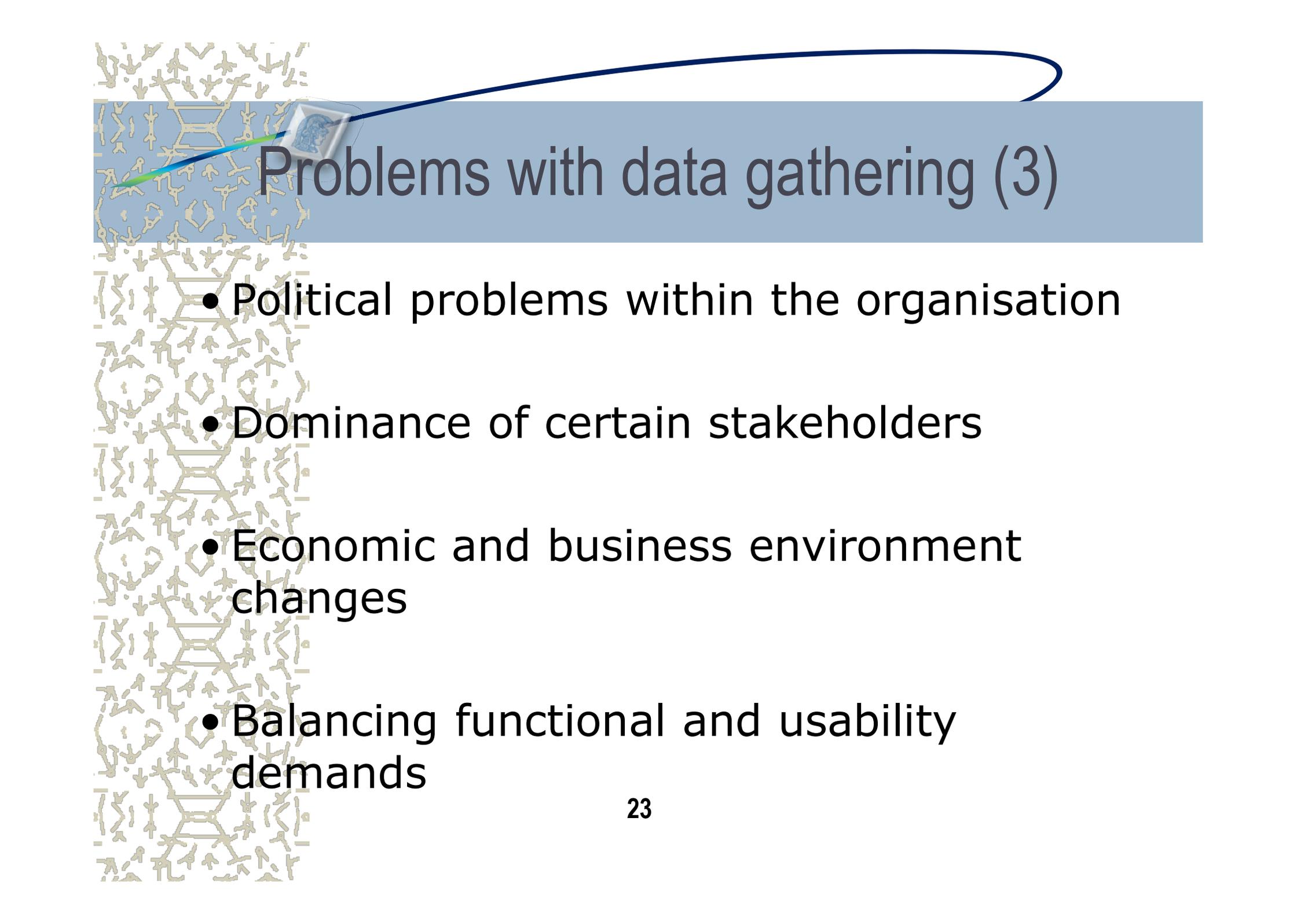
Problems with data gathering (3)

- Political problems within the organisation

- Dominance of certain stakeholders

- Economic and business environment changes

- Balancing functional and usability demands

23
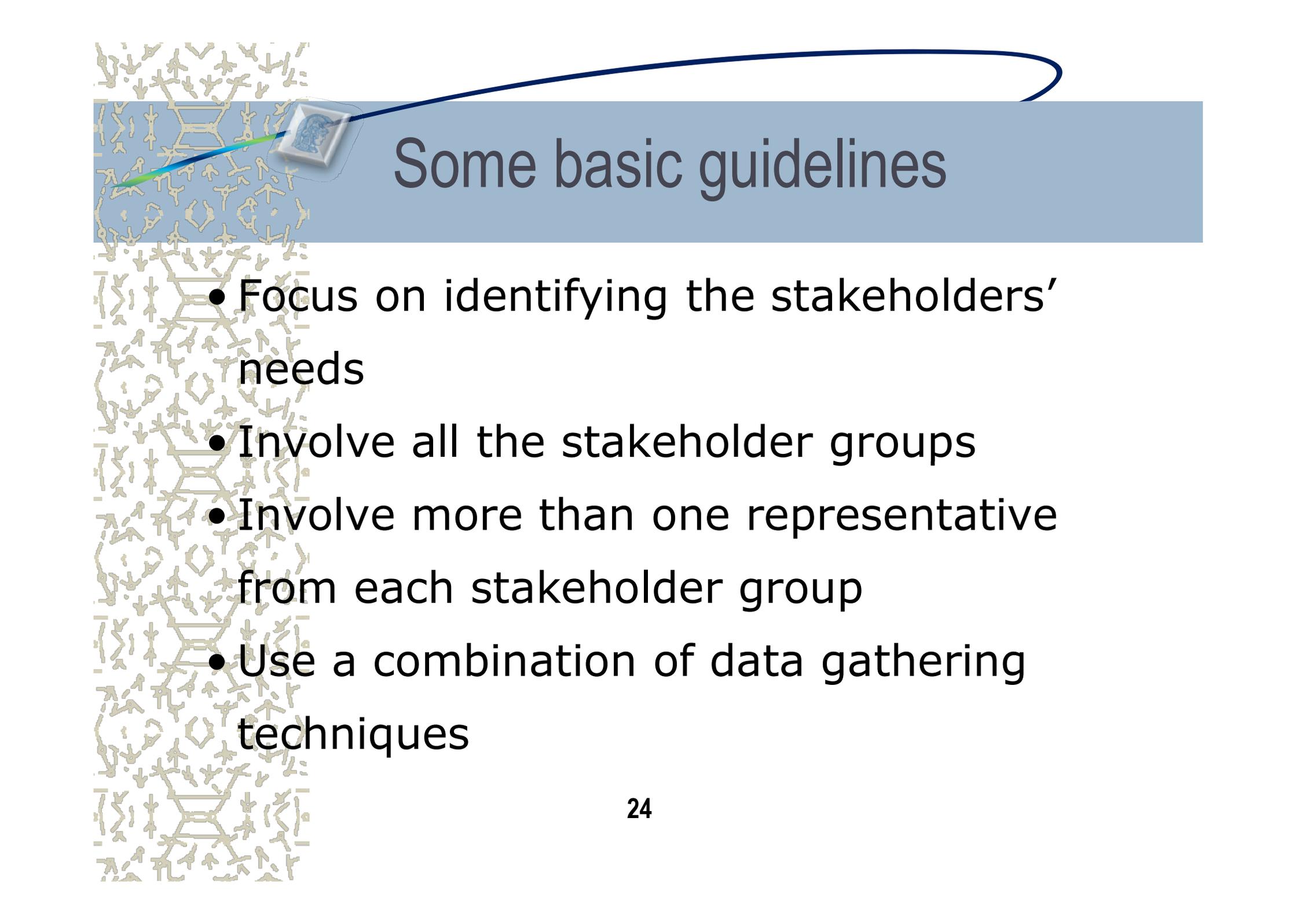
Some basic guidelines

- 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
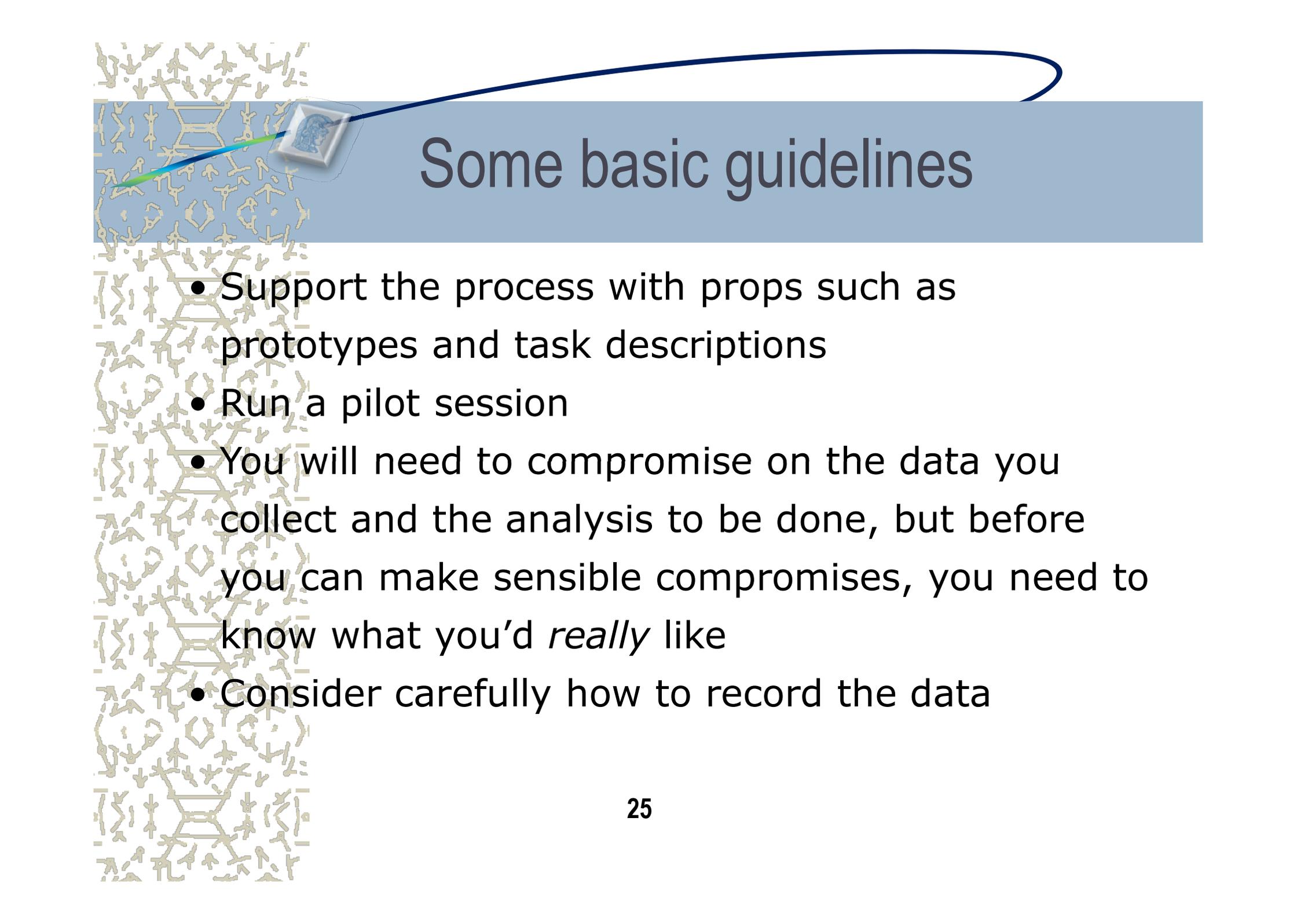
Some basic guidelines

- 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
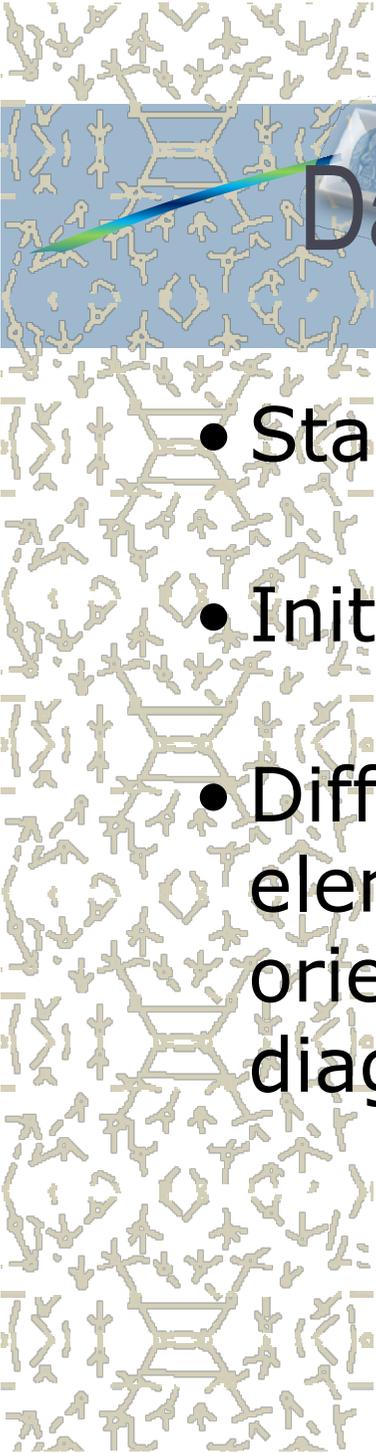
Data interpretation and analysis

- 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
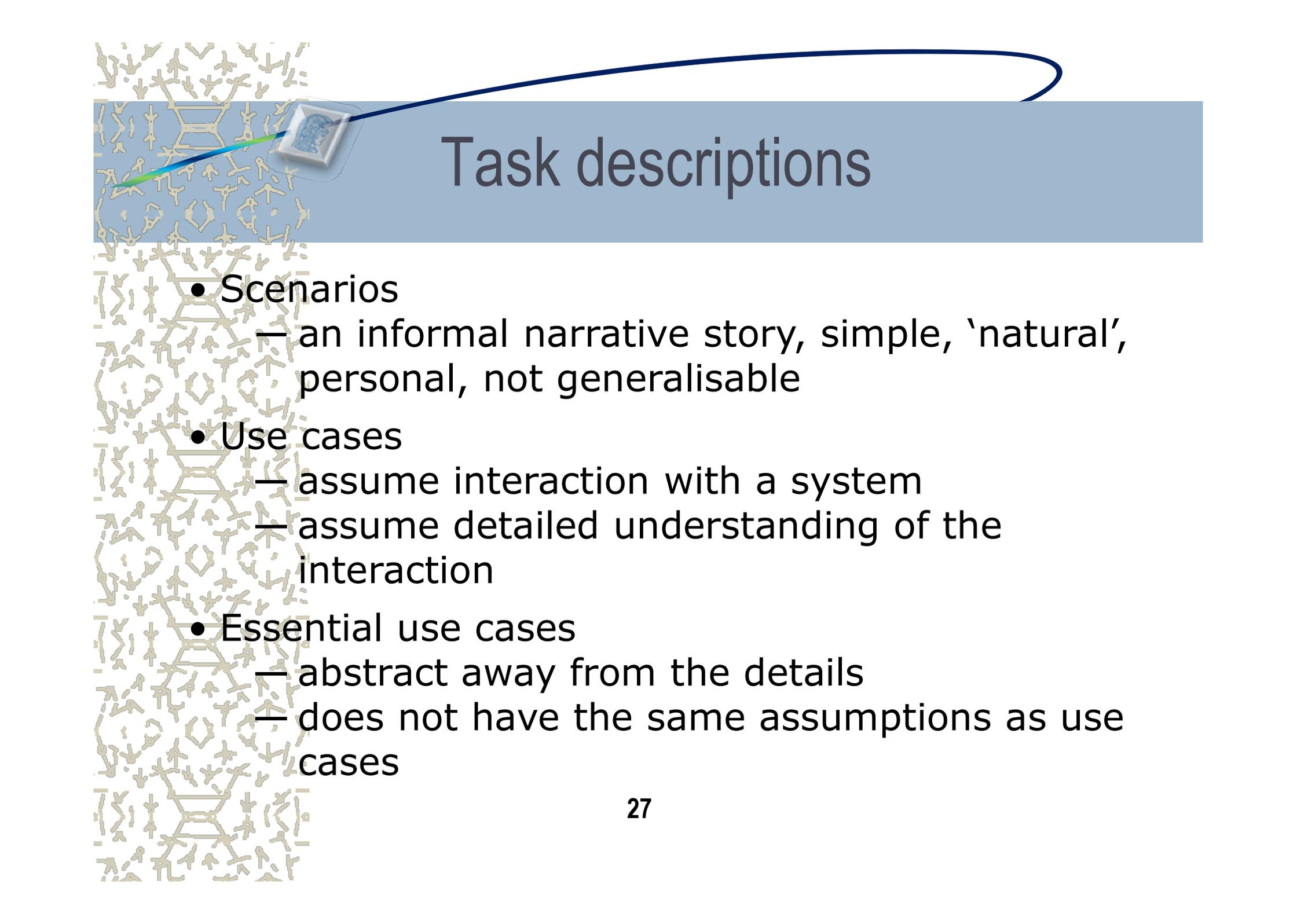
Task descriptions

- 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
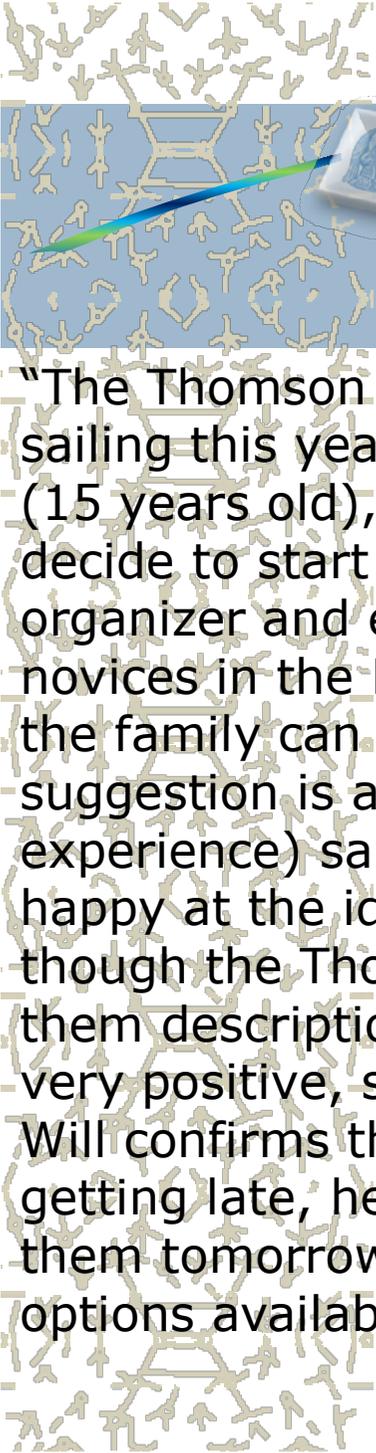
Scenario for travel organizer

"The Thomson family enjoy outdoor activities and want to try their hand at sailing this year. There are four family members: Sky (10 years old), Eamonn (15 years old), Claire (35), and Will (40). One evening after dinner they decide to start exploring the possibilities. They all gather around the travel organizer and enter their initial set of requirements – a sailing trip for four novices in the Mediterranean. The console is designed so that all members of the family can interact easily and comfortably with it. The system's initial suggestion is a flotilla, where several crews (with various levels of experience) sail together on separate boats. Sky and Eamonn aren't very happy at the idea of going on vacation with a group of other people, even though the Thomsons would have their own boat. The travel organizer shows them descriptions of flotillas from other children their ages and they are all very positive, so eventually, everyone agrees to explore flotilla opportunities. Will confirms this recommendation and asks for detailed options. As it's getting late, he asks for the details to be printed so everyone can consider them tomorrow. The travel organizer prints out a summary of the different options available."

"Dan enjoys taking part in virtual chat environments. Late one night, he is in conversation with someone who recommends that he go and see the latest James Bond movie which has just come out. He decides to use the internet to obtain some tickets for the following weekend. At the cinema website he looks for the film titles currently showing. The structure of the site is quite clear, and it's possible to go straight to the information about films and showing times. The James Bond movie is indeed showing. From this page, he can indicate the time of his choice and order the tickets. He chooses the 7pm performance, but the system tells him that this is fully-booked and offers him alternatives: the 5.30pm and the 8pm showings both have available seats. The system displays the seating plan for the cinema which shows the available seats for each showing, and how much each costs. Dan then chooses the seats and showing time that he wants and confirms the booking. Next year, when he has his own bank account, he'll be able to pay for tickets online too and they can be posted to him, but for now he must collect the tickets from the box office and pay for them an hour before the film starts. As he is partially deaf, he needs to double-check that the cinema is equipped with suitable sound amplification technology that links in to his hearing aid. Having completed his order, he returns to chatting with his friends. "

Use case for travel organizer

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.

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 travel organizer
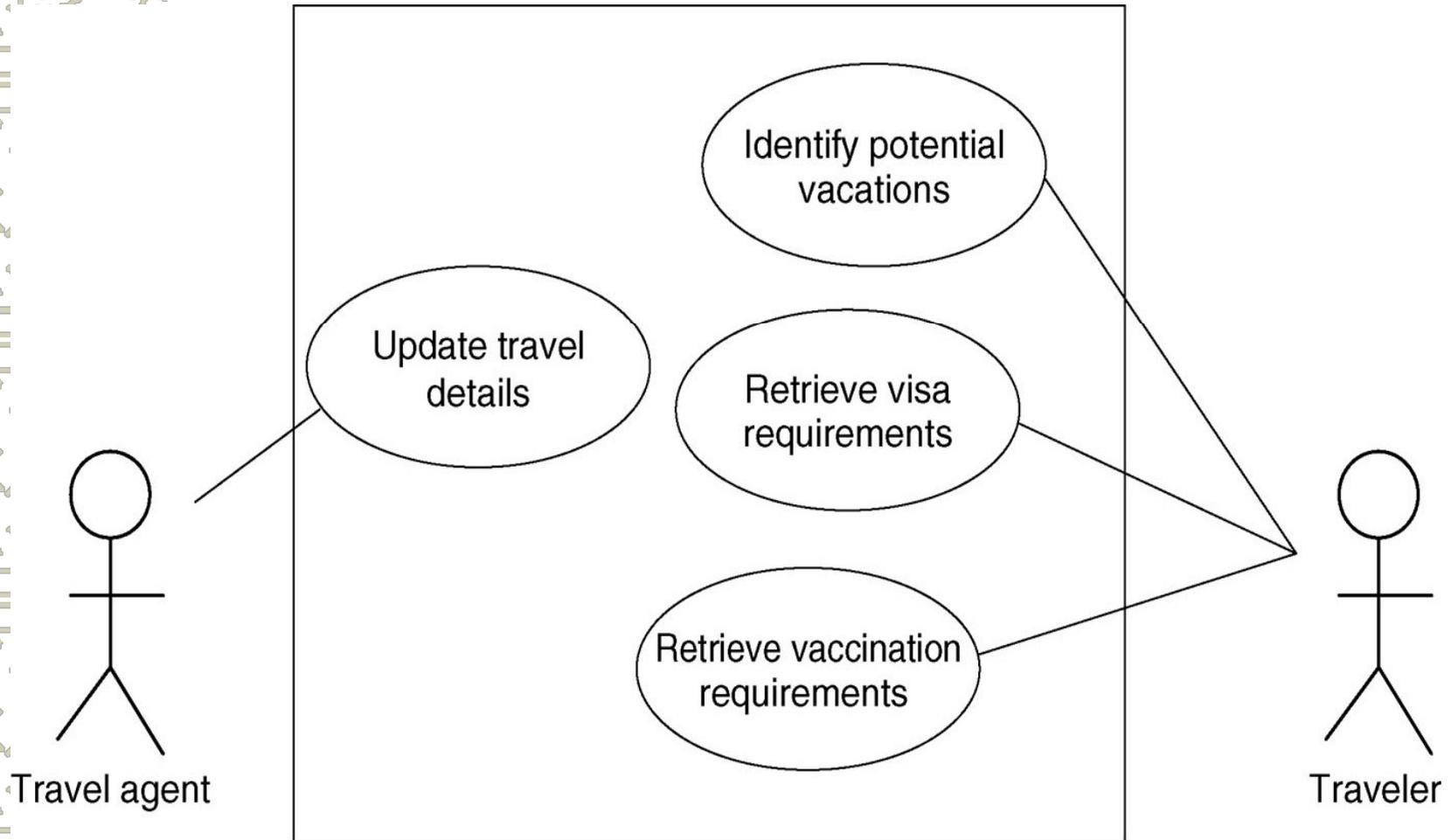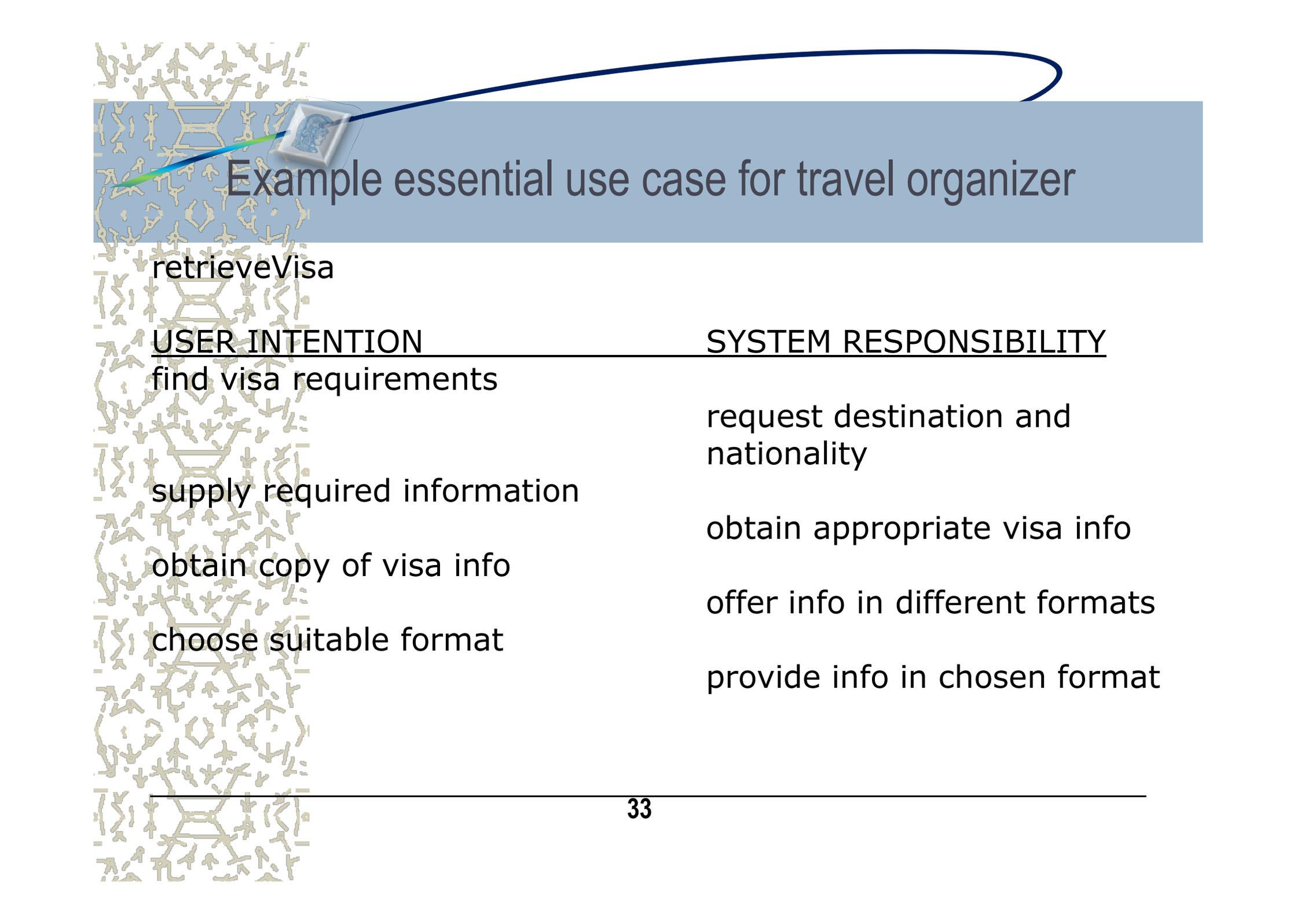
Example essential use case for travel organizer

retrieveVisa

USER INTENTION	SYSTEM RESPONSIBILITY
find visa requirements	
	request destination and nationality
supply required information	
	obtain appropriate visa info
obtain copy of visa info	
	offer info in different formats
choose suitable format	
	provide info in chosen format
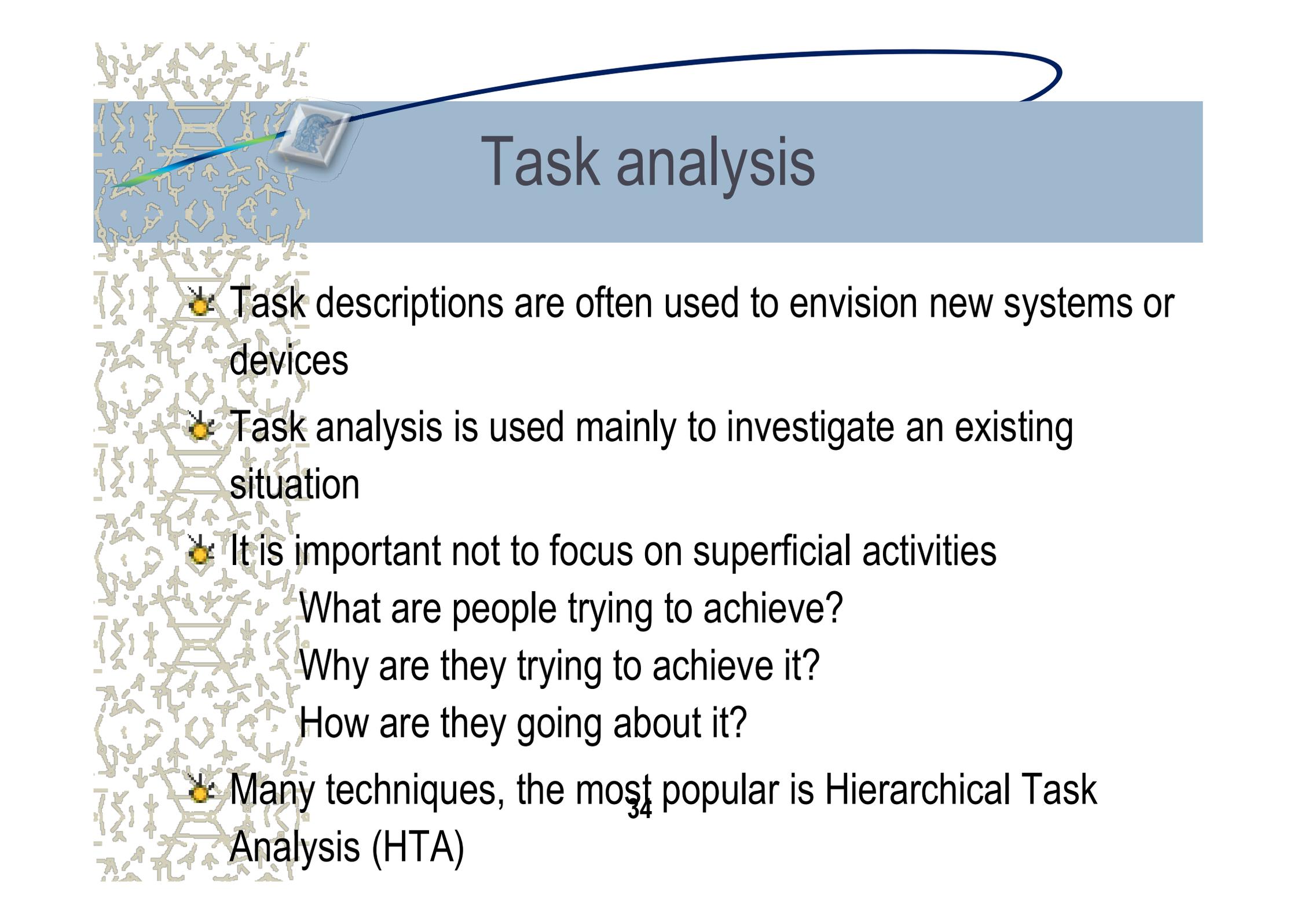
Task analysis

- 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)
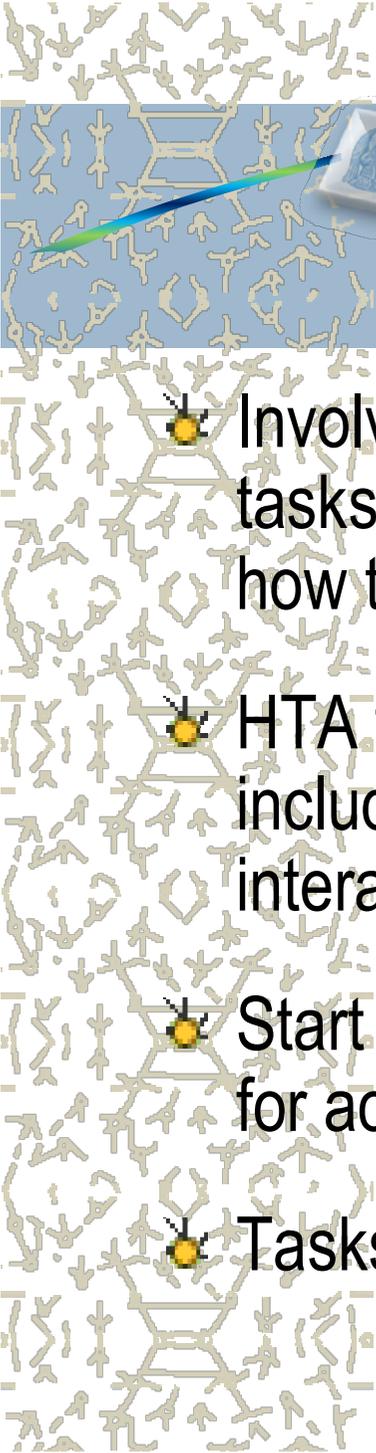
Hierarchical Task Analysis

- 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
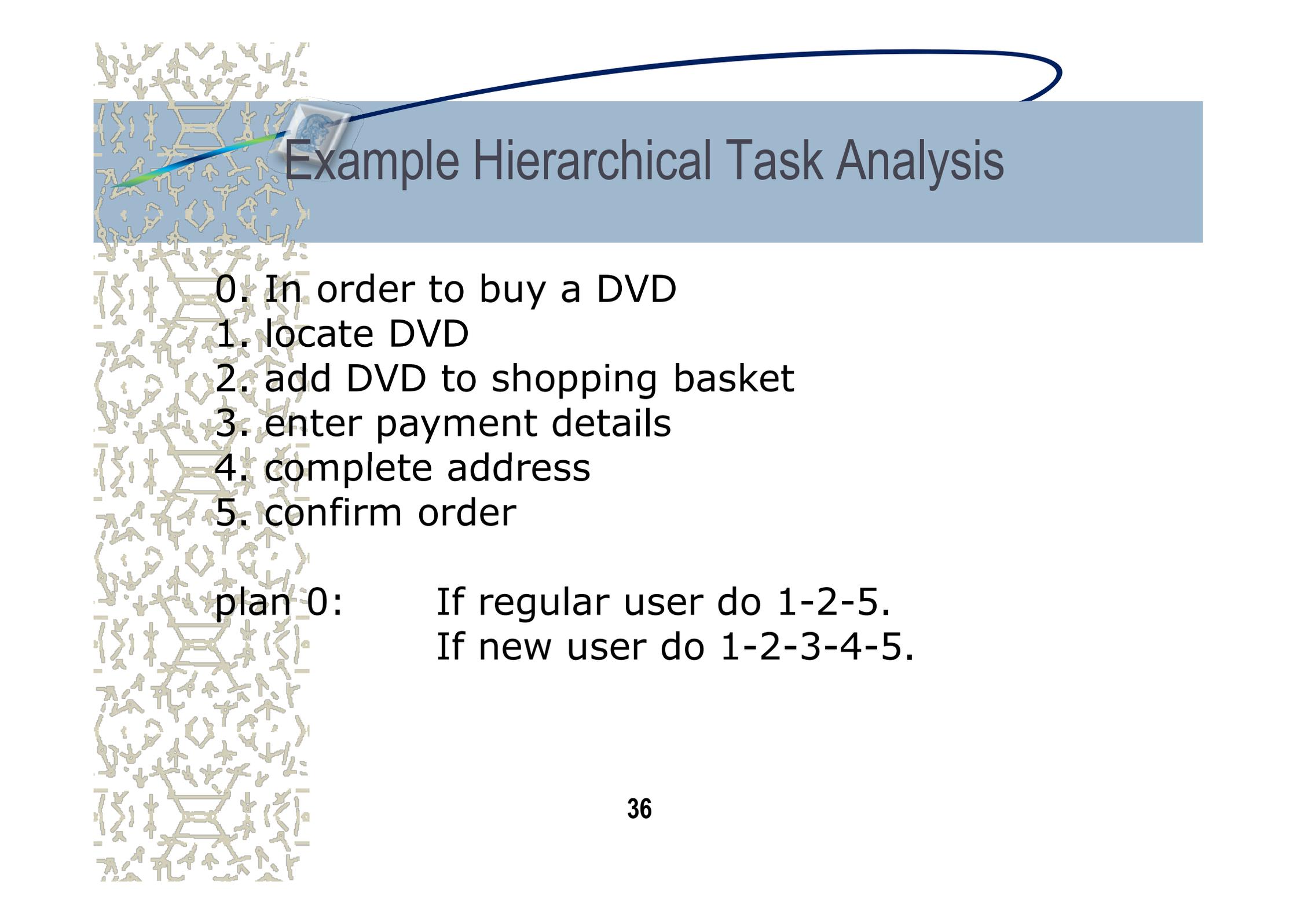
Example Hierarchical Task Analysis

0. In order to buy a DVD
1. locate DVD
2. add DVD to shopping basket
3. enter payment details
4. complete address
5. confirm order

plan 0:   If regular user do 1-2-5.
             If new user do 1-2-3-4-5.

Example Hierarchical Task Analysis (graphical)

```
                        ┌─────────────────┐
                        │ Buy DVD         │
                        │                 │
                        │              0  │
                        └────────┬────────┘
                                 │
                                 │    plan 0:
                                 │    if regular user do 1-2-5.
                                 │    if new user do 1-2-3-4-5.
     ┌──────────┬────────────────┼──────────────────┬──────────────┐
┌────────┐ ┌──────────────┐ ┌──────────────┐ ┌────────────┐ ┌──────────┐
│ locate │ │ add DVD to   │ │ enter payment│ │ complete   │ │ confirm  │
│        │ │ shopping     │ │ details      │ │ address    │ │ order    │
│        │ │ basket       │ │              │ │            │ │          │
│     1  │ │           2  │ │           3  │ │         4  │ │       5  │
└────────┘ └──────────────┘ └──────────────┘ └────────────┘ └──────────┘
```
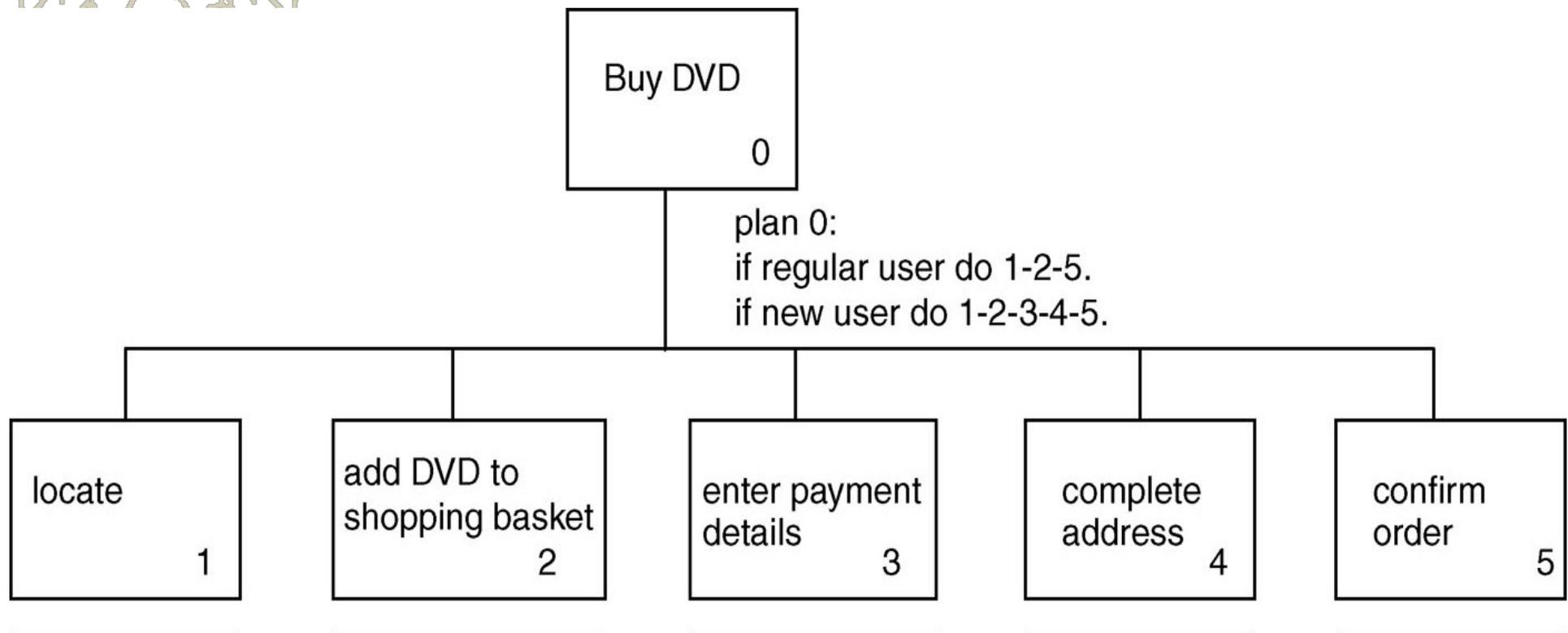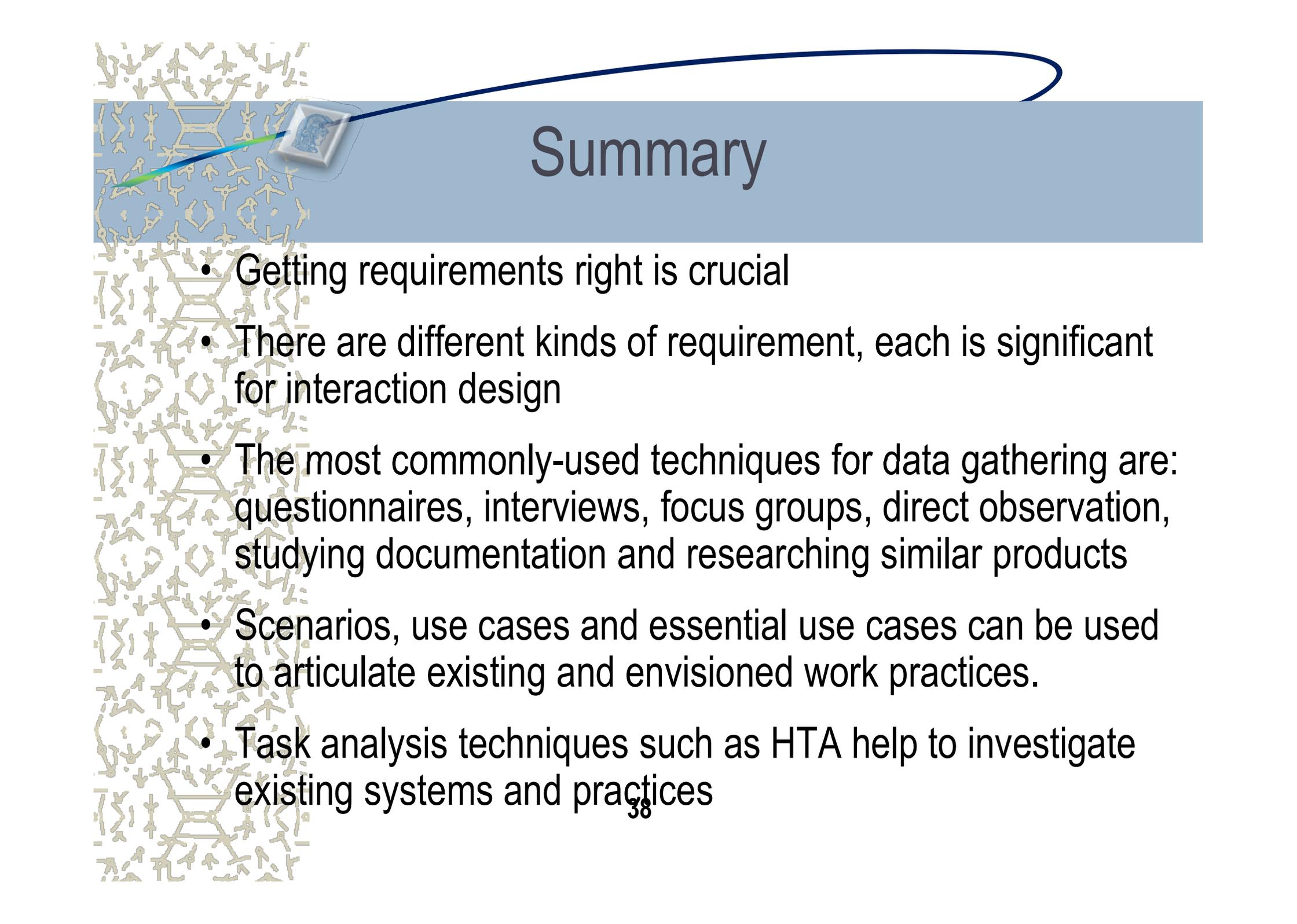
Summary

- 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