



# Unit 4.

# Requirements Analysis

Interfaces Persona Computador  
Depto. Sistemas Informáticos y Computación. UPV

# Unit Goals

- ▶ Learn about User-Centered Design
- ▶ Know the process of requirements gathering and formal specification
- ▶ Study different techniques used to gather information
  - about the users,
  - about the tasks they perform,
  - and the environment where they are performed.

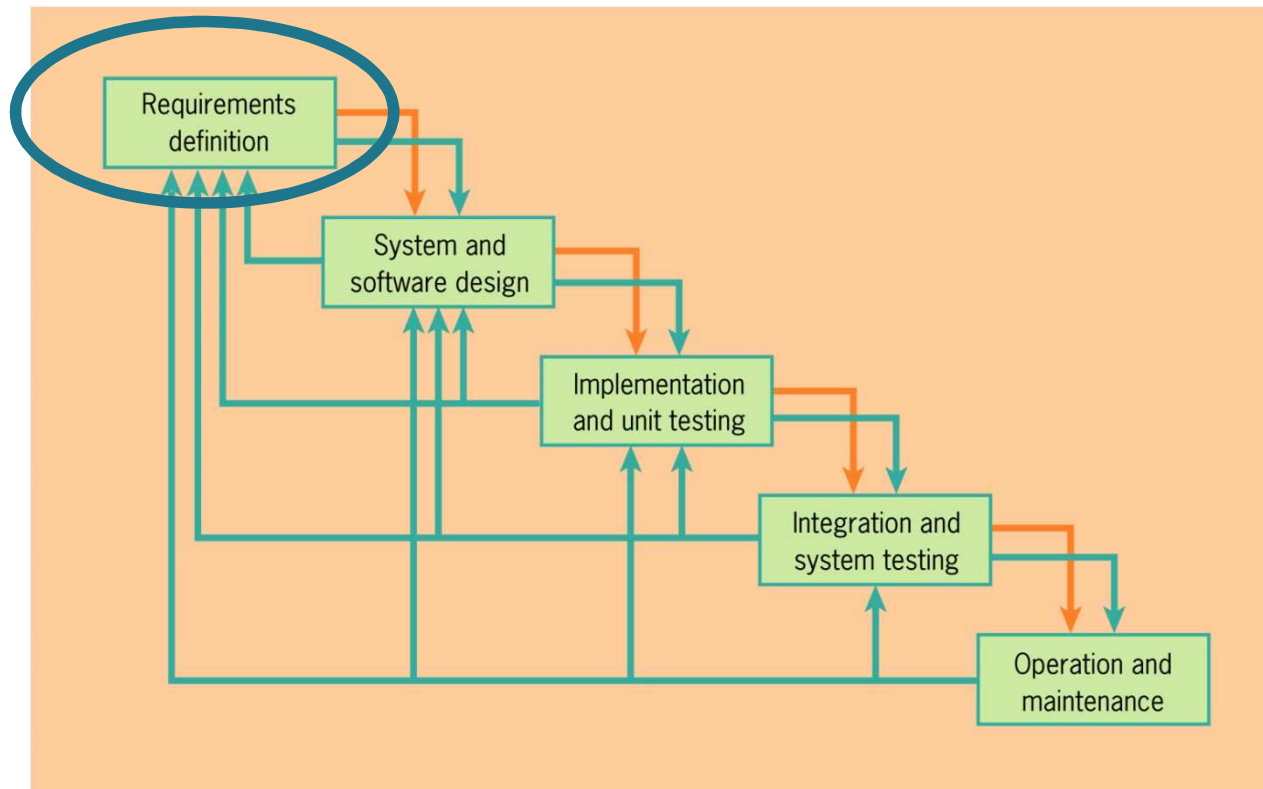


# Overview

- ▶ Introduction
- ▶ User-Centered Design
- ▶ Requirement Analysis
- ▶ Describing users
  - Ethnographic Studies
  - User Studies
  - Personas
- ▶ Describing Tasks
  - Task scenario
  - Concrete use cases
  - Hierarchical Task Analysis
  - Storyboards

# Introduction

- ▶ Remember: the classical life cycle



# User-Centered Design

- ▶ Involve users throughout the design and development process
- ▶ Focuses on understanding
  - the users,
  - the task they perform with the system and
  - the environment in which the system will be used
- ▶ Who are the users?
  - Customers (pay and specify the system), other people in the organization with interest in the development, end users (who actually interact with the system)
    - All of them are referred to as the stakeholders
  - Distinguish between primary users (e.g. the librarian or the travel agent) and secondary users (the person borrowing a book or reserving a flight)



# User-Centered Design

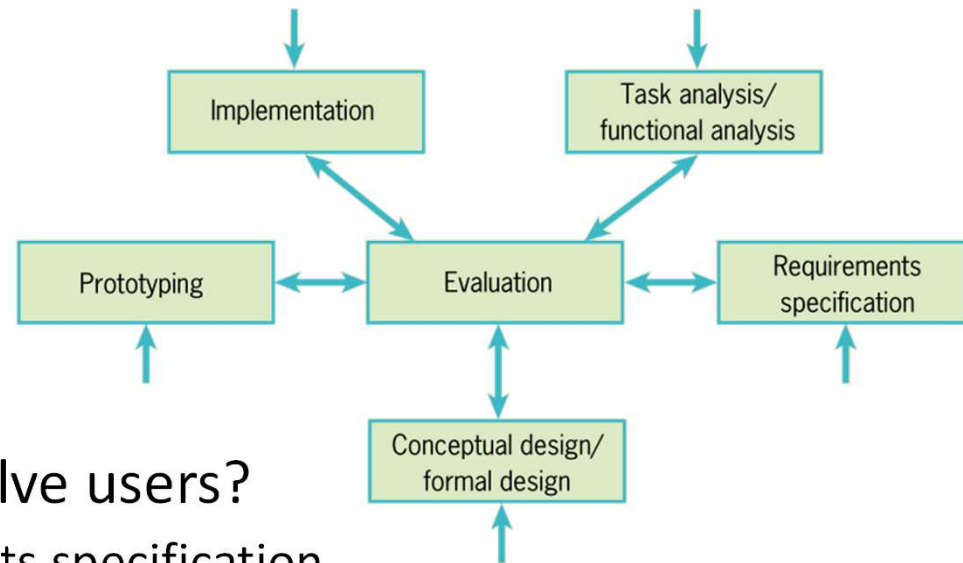
- ▶ Principles of human-centered design:
  - The active involvement of users
  - Appropriate allocation of functions between user and system
  - The iteration of design solutions
  - Multidisciplinary design teams
- ▶ Activities in human-centered design:
  1. Understand and specify the context of use
  2. Specify the user and organizational requirements
  3. Produce design solutions (prototypes)
  4. Evaluate designs with users against requirements





# User-Centered Design

- ▶ The star life cycle



- ▶ When and how to involve users?

- Early: during requirements specification
- During prototyping, to test design and options
- Just before delivery of the product
- During training, after delivery of the system

- ▶ The later in the process, the less changes can be made

# Requirements Analysis

## ▶ Requirement

- A statement about an intended product that specifies what it should do or how it should perform
  - Requirements can be defined with different levels of abstraction
  - Generally they are obtained from observation, interviews or questionnaires
  - They usually describe what the system should do, not how it should do it

- ▶ “A user should be able to obtain the invoice of her purchase. The invoice will be downloadable in pdf format.”





# Requirements Analysis

- ▶ Two types of requirements
  - Functional: what the system should do
    - “A word processor should support formatting at the document, page, word and letter levels”
    - “There should be at least 20 typefaces, each with bold, italic and standard faces”
  - Non-functional: constraints on the system and its development
    - “It should run on Windows, Mac and Linux”
    - “The webpage should load in less than 5 s”
    - “The product must be delivered in less than 6 months”
    - Meet standards of quality, design, security, accessibility, usability...

# Requirements Analysis

- ▶ The Software Requirements Specification is the result of the requirements analysis phase
- ▶ It is a document that gathers all the requirements in a structured template
- ▶ Different standards:
  - IEEE 29148-2011 - Systems and software engineering — Life cycle processes — Requirements engineering. 2011
    - This standard replaces IEEE 830-1998, IEEE 1233-1998, IEEE 1362-1998
  - Volere (next slide)



# Requirements Analysis

## PROJECT DRIVERS

1. The Purpose of the Product
2. The Stakeholders

## PROJECT CONSTRAINTS

3. Mandated Constraints
4. Naming Conventions and Definitions
5. Relevant Facts and Assumptions

## FUNCTIONAL REQUIREMENTS

6. The Scope of the Work
7. Business Data Model and Data Dictionary
8. The Scope of the Product
9. Functional and Data Requirements

## NON-FUNCTIONAL REQUIREMENTS

10. Look and Feel Requirements
11. Usability and Humanity Requirements
12. Performance Requirements

13. Operational and Environmental Requirements
14. Maintainability and Support Requirements
15. Security Requirements
16. Cultural and Political Requirements
17. Legal Requirements

## PROJECT ISSUES

18. Open Issues
19. Off-the-Shelf Solutions
20. New Problems
21. Tasks
22. Migration to the New Product
23. Risks
24. Costs
25. User Documentation and Training
26. Waiting Room
27. Ideas for Solutions

# Req. Analysis. Gathering information

## Ethnographic Studies

### ► Ethnographic Studies

- Observe users while they are doing real work in their real working environment or using a home system in their homes
- Find what users do, but also what users like/dislike



Bronislaw Malinowski with natives on Trobriand Islands in 1918  
[wikipedia.org](https://www.wikipedia.org)

# Req. Analysis. Gathering information

## Ethnographic Studies

### ► Types of observation:

#### ◦ Direct observation

- Field studies: the observer takes notes about interesting behaviors in the work place/home
- Controlled studies: the user interacts with the system in a controlled environment (typically a usability laboratory)
- Pros: easy, produces interesting data
- Cons: only a single pass at the information gathering, hard to record every aspect of the activity, what is not recorded is lost, it is obtrusive and can alter user's behavior and performance

#### ◦ Indirect observation

- Record video, keystrokes, mouse movements, etc.
- Pros: the whole interaction is captured, more objective
- Cons: analysis of data is time consuming, some users may be intimidated

# Req. Analysis. Gathering information

## Ethnographic Studies

- ▶ Observation in the work place
  - Work organization: how it is currently done
  - Initial capture of actors and tasks
  - Write the conditions of the position or the work place
  - Catalogue all relevant artifacts in the work environment
  - The artifacts can be both physical or conceptual:
    - Brochures (corporate identity)
    - Work procedures
    - Social environment
    - ...



# Req. Analysis. Gathering information

## Ethnographic Studies

- ▶ Results of an ethnographic study
  - Activity and job descriptions
  - Rules and procedures said to govern particular activities
  - Descriptions of activities observed
  - Recordings, photographs, videos...
  - Informal interviews with participants explaining the activities
  - Diagrams, workflow, order and sequence of tasks
  - ...



# Req. Analysis. Gathering information

## User Studies

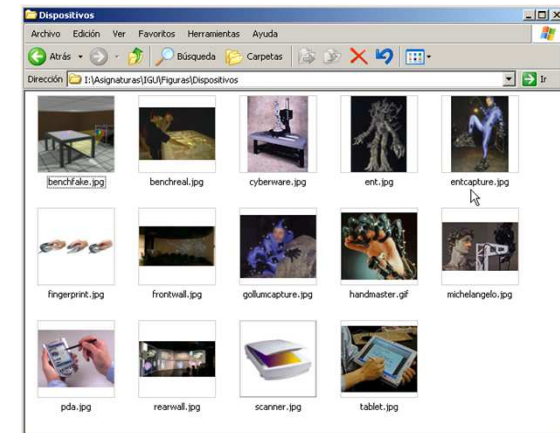
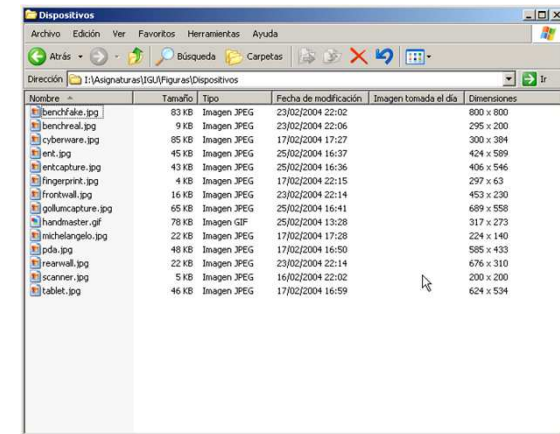
- ▶ Finding about the users and the domain
  - Describe and classify users depending on different characteristics
    - Age, sex, culture, physical abilities and disabilities, educational background, computer/IT experience, motivation, attitude
  - Interview real users or, if not possible, domain experts, managers, work supervisors... who may know about them
- ▶ Extract user groups, and focus your design on those groups
  - A person in one group will typically interact with the system differently than a person in other group



# Req. Analysis. Gathering information

## User Studies

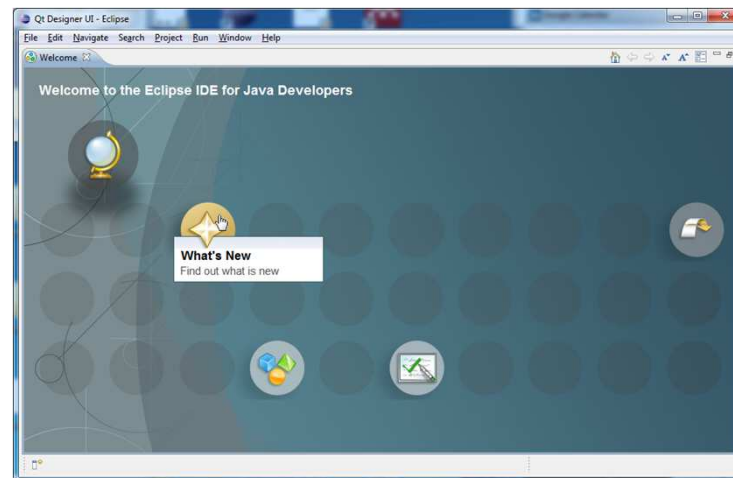
- ▶ Novice or first-time users
  - Lack of previous experience may cause anxiety
  - Restrict the vocabulary, use online help, dialog boxes, give positive feedback and specific, constructive error messages
- ▶ Knowledgeable intermittent users
  - Problems retaining the structure of menus or the location of features
  - Use structure, consistent terminology and sequence of actions. Emphasize recognition rather than recall. Provide guides to frequent activities. Use contextual help and undo tools
- ▶ Expert frequent users
  - They demand rapid response times, brief and nondistracting feedback and shortcuts.
  - Use of macros and accelerators.



# Req. Analysis. Gathering information

## User Studies

- ▶ Designing a user interface for one type of user is easy. Design a UI for different levels of experience is harder (multi-layer)
  - Give the novices a reduced, simplified interface
  - Depending on each user's progression, add higher level tools to perform more complex tasks
  - Apply the multi-layer design to develop the online help, error messages and tutorials
  - Ideally, allow the user to configure the environment: menu contents, feedback level, information density or pace of the interaction



# Req. Analysis. Gathering information

## User Studies

### ► Interviews

- Very important: plan ahead
  - Who to interview, what questions to ask, how long should last...
  - Two types of interview:
    - Structured : predefined set of questions
      - Pros: easier to conduct, easier to analyze
    - Flexible: some preset topics, but no set sequence
      - Pros: less formal, obtain more information and opinions
  - What you should do:
    - Make the person feel comfortable. Some users won't express their real opinions because they think it is their fault, or that their opinions are trivial, or have no importance
    - Perform a small pilot study and record the interviews (with permission)
    - Start with an open question

# Req. Analysis. Gathering information

## User Studies

- ▶ Interviews: what not to ask?
  - Asking leading questions (would you like a less cluttered interface?, is the daily update an important feature to you?)
  - What they would do/like/want in hypothetical scenarios?
  - How often they do things (how often do you exercise in a typical week?)
  - How much they like things on an absolute scale
  - Binary questions





# Req. Analysis. Gathering information

## User Studies

- ▶ And, apart from what people say, pay attention to what people do
- ▶ People
  - (usually males) try not to look stupid
  - try to answer what they think we want to hear
  - are not aware of their behavior



# Req. Analysis. Gathering information

## User Studies

### ▶ Questionnaires and Surveys

- They are a set of questions and statements for gathering more precise information. Two types of questions:
  - Closed
    - Yes/no/Don't know
    - Multipoint rating scale between two opposites (for example, very useful... Of no use)
  - Open
    - No predetermined answers
    - The user can respond what she thinks in the provided space
    - Gives more information, but are harder to analyze
- A questionnaire should be:
  - understandable, as short as possible, captures the information you need, provides the option to add additional thoughts

# Req. Analysis. Gathering information

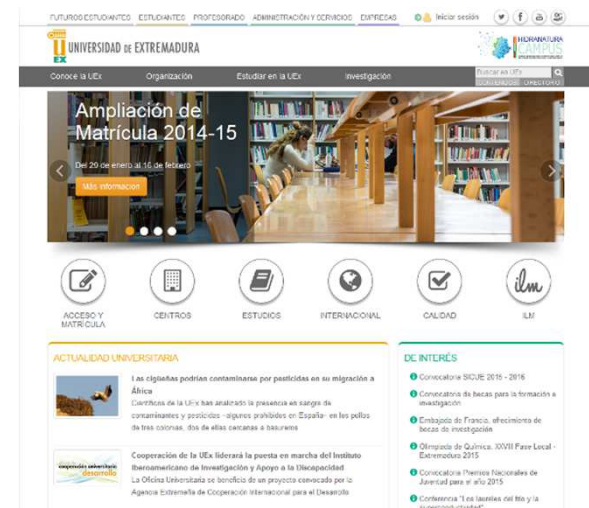
## Studying the Competence

- ▶ Analysis of the competition
  - It is a fast and easy way for establishing a starting point in the design
  - Activities:
    - Make a list of the existing products similar to your idea
    - Create a comparative table with their evaluation
    - Make a presentation, focus group, etc. for reviewing the results
  - Review the previous versions of the application
  - Add to your design good ideas about the usability of existing applications



# Req. Analysis. Gathering information

## Studying the Competence



# Requirement Analysis

## Describing Users

### ▶ Personas


- A model of a person
  - includes the person's motivation, likes and dislikes, intentions, behavior and goals
  - Define her attitudes towards new technologies and how she currently uses the system
- Draw a picture of your persona or use a photo
  - give her a name, age, an occupation, a background, a social situation and goals etc. Give the persona a story to tell
- Knowing what our persona thinks, does, and feel help build empathy
  - so that you can understand the state of mind, emotion, philosophy, beliefs, or point of view of the user
- The persona is built from the results of the real users interviews



# Requirement Analysis

## Describing Users

**Confident learner**



Picture credits –  
[Nerdcorgirl Flickr CC](http://www.flickr.com/photos/nerdcorgirl/)  
<http://www.flickr.com/photos/nerdcorgirl/>

### Samantha Bell

**“I’d love to keep in contact with my friends”**

Sam is about to go abroad for her gap year, so her parents decided to get her a new camera, to make sure she’s able to record everything she gets up to.

She likes the camera as it looks so modern, and it’s able to do so much more than a lot of her friends’ cameras.

She loves being in contact with people all the time, and finds it’s a great way to kill time like when waiting for the bus. She uses a lot of the more advanced features – panoramic shots, online upload and .

When she encounters a problem she ignores it most of the time - she’s not sure if she even got a manual with the camera. When she has trouble she can’t ignore she speaks to her friends, or goes into a camera store – she wants to be talked through the problem.

### First time user

Female, 27 year old, single  
Student

Sam prefers to learn how to things by trying things out by herself. She isn’t worried about ‘breaking’ anything. If she does need help she would prefer to not to refer to a manual but “do it herself”.

### Needs

In order of preference:

1. To share pictures with her parents
2. To share her pictures with her friends
3. To share her pictures with people she meets whilst travelling

### Ideal features

- Ability to take pictures
- Ability to upload images to personal site using 3G/Wifi
- Allowing others to access her pictures remotely
- Long battery life
- Ability to name and add comments to uploaded images
- Ability to create several albums, and upload pictures to each

### Frustrations

- Lack of wireless/3G access
- Slow uploads
- Low battery life
- Need to be plugged in to upload images
- Slow shutter speed
- Want to be able to name/add comments to uploaded images
- Getting online is confusing
- Creating new albums

### Key attributes

|            | Low | High |
|------------|-----|------|
| Knowledge  |     | ●    |
| Experience | ●   |      |
| Help use   | ●   |      |
| Confidence |     | ●    |

Webcredible – user experience research & design

March 2010

<http://www.webcredible.com/blog-reports/web-usability/personas.shtml>



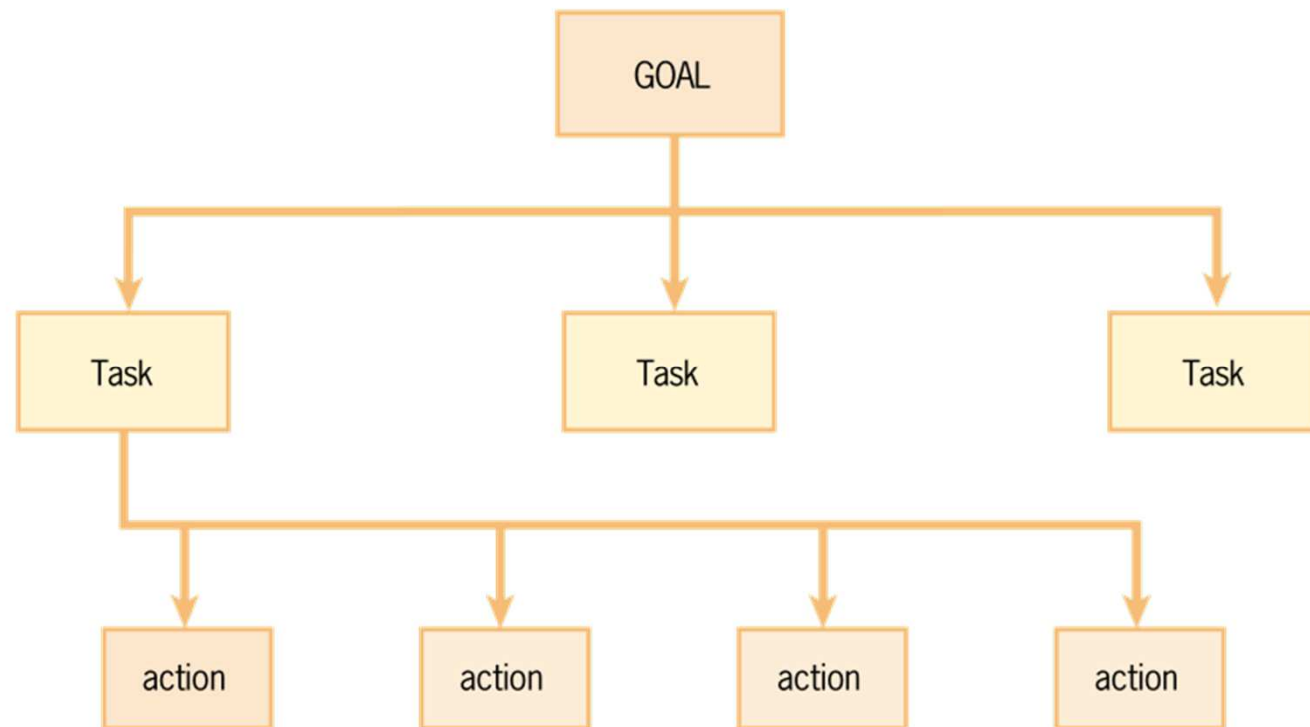
# Tasks Analysis

- ▶ After identifying the users, we must understand their goals when using the system
- ▶ Task analysis is an activity to study what a system must do and the functionality it has to provide to help users achieve their goals
- ▶ Terminology:
  - Goal: end result to be achieved (communicate with a friend in writing)
  - Task: structured set of activities undertaken in some sequence
  - Action: individual operation or step of a task



# Tasks Analysis

- ▶ Relationship between a goal, a task and an action



# Task Analysis

## Gathering Information

- ▶ Information about the users' tasks can be obtained by interviews, observation and study of documentation
- ▶ But also:
  - Study the most common errors users make in the current system
  - Find work arounds: ways users find of doing tasks when the UI does not support the task (usually complex task sequences)
  - Find artifacts: notes, cheat sheets, sticky notes, overused reference manuals, annotated pages, paper forms...
- ▶ Ask the user specifically for them, because may be they are so used to them, that they don't realize there is a problem

# Tasks Analysis

## ▶ Task Characteristics

- The extent to which tasks vary from one occasion to another
- Whether tasks will be carried out regularly, infrequently, or only once
- The knowledge and kinds of skill required to perform tasks
- How much the work is affected by changes in the environment
- Whether time is critical for the work
- Whether there are safety hazards
- Whether the user will do the work alone or with others
- Whether the user will normally be switching between several tasks



# Tasks Analysis

- ▶ The sequence of tasks to achieve a goal can change from user to user
  - The UI should not impose a sequence of tasks unless there is a reason



# Describing Tasks

## Task Scenario

- ▶ Task scenario: is a narrative description of a task, as is currently being done
  - They tell a story about the use of a system
  - Personalized: describe a specific instance and situation
  - Detailed: describe step by step the procedure followed by the user to get a task done, and the features and behavior of the system
  - Includes problems and difficulties
  - They should be evaluated by the users to check that they describe faithfully the task





# Describing Tasks

## Task Scenario

### **Task scenario. Search and request resource**

Julia, a lecturer in the department, is looking for a particular CD-ROM containing examples and exercises on Object Oriented Analysis and Design. She knows that Tom, another lecturer, mainly teaches Object Oriented Analysis and Design so she knocks on his door. Unfortunately he is not there, so she leaves a note on his door. Later he returns and searches for her, finding her in the coffee bar. He tells Julia that Geoff has the CD-ROM. Unfortunately Geoff is on leave, so Julia telephones him and he promises to post it to her.

### **Task scenario. View updates and request resource**

Mark has recently returned from six months of study leave and wants to find out what books other members of the department have bought since he left. To do this he telephones everyone in the department and arranges an appointment. He has to do this because everyone is at the university at different times. He then meets everyone individually and checks through their bookcases, asking to borrow books that interest him. He only asks for one book at a time, as he is a slow reader!

# Describing Tasks

## Task Scenario

- ▶ Use scenario: they are similar to task scenario, but they describe the anticipated use of the system

### **Use scenario. Search and request resource**

Julia is looking for a particular CD-ROM containing examples and exercises on Object Oriented Analysis and Design. She accesses the digital library from home and types in the key phrase 'Object Oriented Analysis'. The system retrieves one result. Geoff owns the appropriate CD-ROM. Julia then sends an e-mail to Geoff, asking to borrow the CD-ROM.

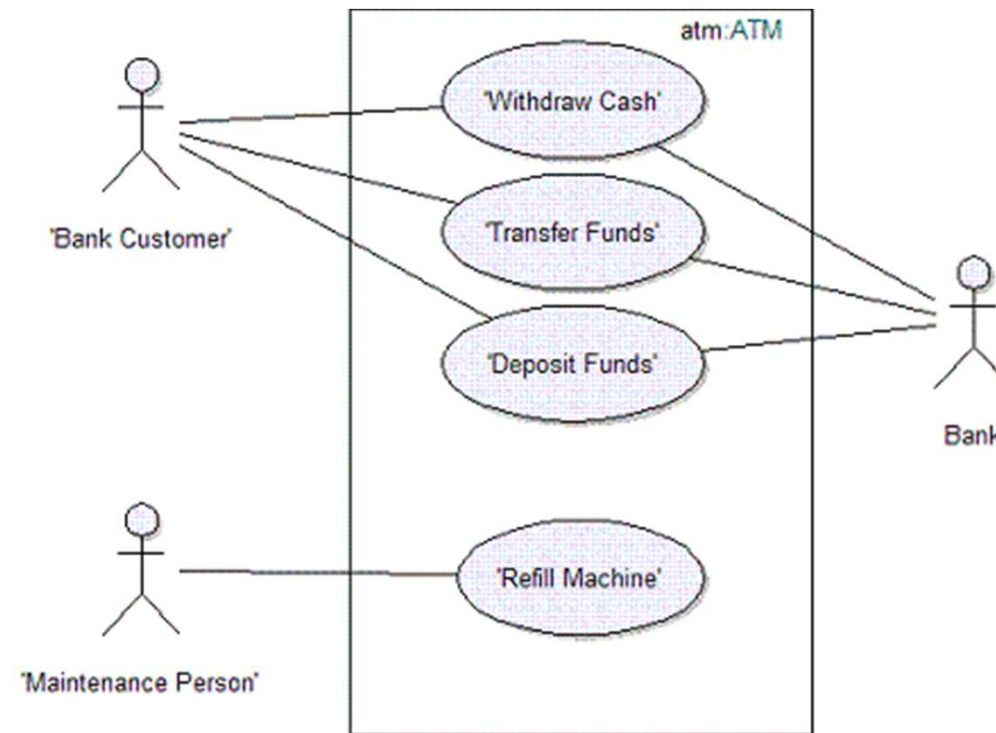
### **Use scenario. View updates and request resource**

Mark has recently returned from study leave and wants to find out what are the latest additions to the digital library. He selects 'check updates', identifies the books he is interested in, and sends an e-mail to the owner of the one that interests him most.

# Describing Tasks

- ▶ Use cases also focus on user goals, but the emphasis is on a user-system interaction rather than the user's task itself
  - The users are called actors, and other systems that interact with the system being described can also be actors
  - For example: withdraw money from an ATM
    - Prerequisite: the user is identified
    - Steps
      - The system shows the available operations
      - The user selects “withdraw money”
      - The system asks the user if he wants to use credit or debit
      - The user selects an option
      - ...

# Describing Tasks



<http://epf.eclipse.org/wikis/openup/>

# Describing Tasks

## Concrete use cases

- ▶ Concrete use cases
  - They are similar to task scenarios, but they are not personalized (they are a little bit more generic)
  - They can be written in two columns, one column for user actions and another column for system responses.





# Describing Tasks

## Concrete use cases

| User action   | System response  |
|---|--|
| The academic enters one or more of the search parameters for the CD-ROM: title, year and platform | The system displays the search results   |
| The academic selects a search result  | The system displays the full details of the CD-ROM and the contact details for its owner who is a research student |
| The academic chooses the e-mail address   | The system displays a message area   |
| The academic writes and sends the e-mail request  | The system confirms the sending of the request   |



# Describing Tasks

## Hierarchical Task Analysis

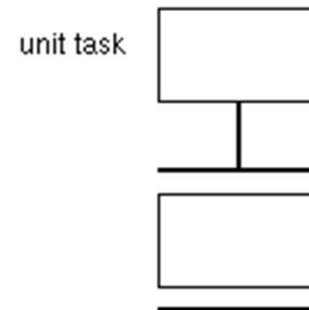
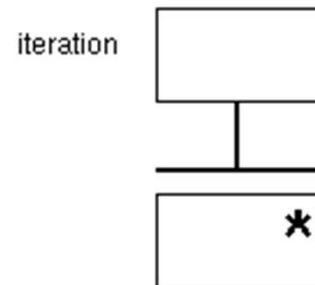
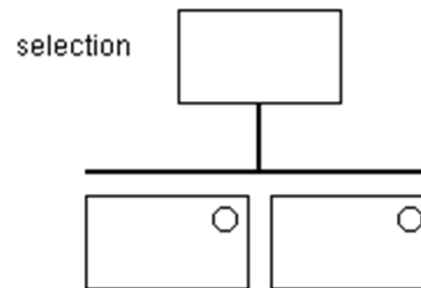
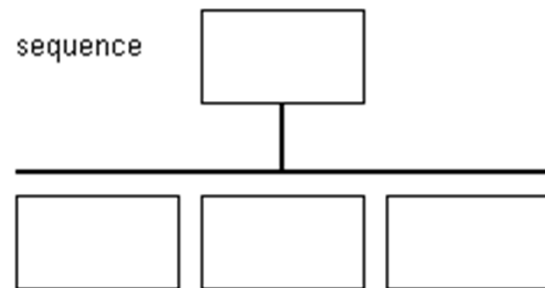
- ▶ Hierarchical Task Analysis
  - Involves breaking a task down into subtasks and then into sub-subtasks and so on.
  - The subtasks are then grouped together as plans that specify how the tasks are performed in an actual situation
  - Focuses on the physical and observable actions that are performed
    - including actions not related to software or an interactive product at all
  - The starting point is a user goal



# Describing Tasks

## Hierarchical Task Analysis

### Notation (actions)



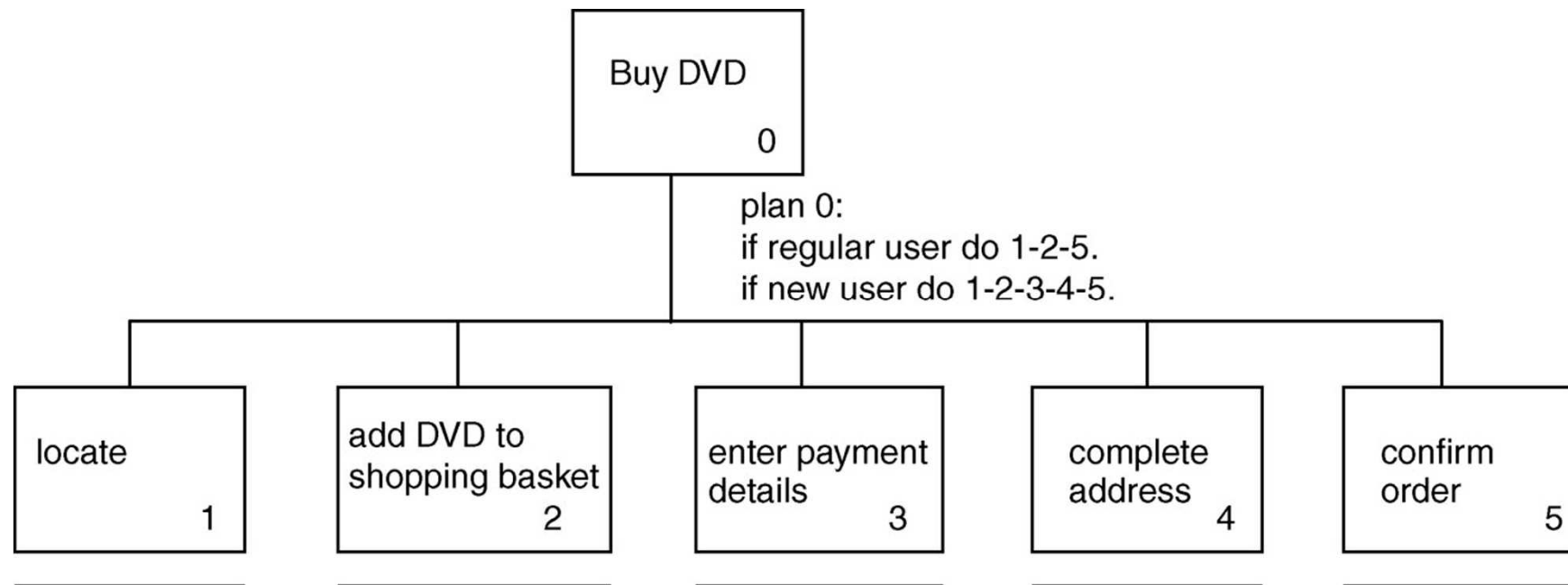
### Plan

```
plan 0
do 1
if it is a work day
then do entonces hacer 4-5-6-7
else do 10
...
```

# Describing Tasks

## Hierarchical Task Analysis

### ► Buy a DVD



# Describing Tasks

## Task Environment

- ▶ The Physical Environment
  - Lighting, temperature, noise, dirt, dust...
  - Physical design and layout
- ▶ The Safety Environment
  - Special clothing, safety critical, pollution or other hazards...
- ▶ The Social Environment
  - Stress, cooperation/work alone, dependencies, distractions, social hierarchy...
- ▶ The Organizational Environment
  - Mission and aims, purposes, working factors (hours, group, management, attitudes, flexibility, performance monitoring...)
- ▶ The User Support Environment



# Describing Tasks

## Storyboards

- ▶ Storyboards are a sequential series of illustrations, stills, rough sketches and/or captions (sometimes resembling a comic or cartoon strip) of events, as seen through the camera lens, that provide a synopsis for a complex scene with its action and characters



<http://pbcbstudios.tumblr.com/image/69033347508>



[digititles.com/  
animation/brave-2012](http://digititles.com/animation/brave-2012)

# Describing Tasks

## Storyboards

- ▶ Storyboards are used to describe tasks
  - They are usually hand drawn
  - They show the flow of interaction
  - They show people doing a task to achieve a goal
  - Do not spend too much time drawing





# Describing Tasks

## Storyboards

- ▶ Storyboards should convey
  - Setting
    - People involved
    - Environment
    - Task being accomplished
  - Sequence:
    - What steps are involved
    - What leads someone to use the app?
  - Satisfaction
    - What motivates people to use this system?
    - What does it enable people to accomplish?
    - What need does the system fill?

# Describing Tasks

## Storyboards

- ▶ Benefits of Storyboarding
  - Helps emphasize how an interface accomplishes a task
  - Avoids commitment to a particular user interface (no menus, buttons, etc.)
  - Helps get all the stakeholders agree in terms of the goal



# References

- ▶ D. Stone, C. Jarrett, M. Woodroffe. User Interface Design and Evaluation. Morgan Kaufmann, 2005.
  - Chapters 1-4
- ▶ A. Dix, J. Finlay, G. Abowd, R. Beale. Human-Computer Interaction. Prentice Hall, 2004.
  - Chapter 15
- ▶ Scott Klemmer. Human-Computer Interaction course in Coursera:
  - Interviewing: <https://www.coursera.org/learn/human-computer-interaction/lecture/itTXx/interviewing>
  - Storyboards. First half of: <https://www.coursera.org/learn/human-computer-interaction/lecture/78yeB/storyboards-paper-prototypes-and-mockups>

