by Nacho Iborra

# Development environments
# Block 1
# Unit 4: Requirements specification of an application

## 1.4.1. Starting a software project

When a software development company decides to start a software project, it has to take into account some issues like: where does the idea come from? who will be in charge of the project?

### 1.4.1.1. Origin of the project

Regarding the origin of the idea, it can come from:

- The company itself, as one of its internal developments. In this case, the company becomes its own customer.
- An external entity/company which asks this company to develop the product. In this case, the company acts as a software provider.

In both cases, we must have a clear overview of the product to be developed, and evaluate its viability. This is easier when it comes from external entities, since the agreement can fit the needs of the software company in order to make a profitable product. But, when it is an internal development, as the company has no additional incomes, it is essential to evaluate if this project is really worth it.

### 1.4.1.2. The project manager

Regarding the project manager, we have to take into account some factors, such as:

- This manager should take part in the decision of starting the project
- The chosen candidate should have a proven experience in the business area of the project, besides having good interpersonal relationships and group management.

The project manager must do tasks like:

- Identify the possible risk areas of the project
- Set budgets, calendars and task assignments
- Support and provide communication techniques among the team members.
- Set how to communicate with the customer

## 1.4.2. Information gathering

The communication with the customer is an essential part of the decision-making process, since we can get to know what he really wants. In this process we have two main drawbacks:

- Developers don't know (and they may never get to know) what the customer exactly wants.
- Customers don't know what type of information is relevant for the development team in order to implement the application.

In order to reduce these drawbacks as much as possible, we must:

- Identify the sources of information that are suitable for our purposes in the customer's company
- Ask the appropriate questions and understand the customer needs
- Check with the customer what we have understood about the application requirements
- Synthesize the requirements in a document called *system requirements specification* (SRS).

## 1.4.2.1. Techniques to gather information

There are several techniques that we can apply to gather information from the customer, set the application requirements and evaluate its viability. Some of the most popular (and not necessarily excluding) are:

- **Interviews**: it is maybe the most popular one, although it requires a good level of preparation and experience from the analyst in order to be successful.
- **Joint Application Development (JAD)**: customers and analysts work together in the features that the product must have. This way, the customer is more involved in the development process.
- **Observation**: it consists in analyzing (in the customer's company) how the company works and how the software product may fit in this company.
- **Documentation study**: a study of the official documents of the company, which describe its internal performance
- **Questionnaires**: they are useful to gather information from many people in a short time. It is also useful when these people are very far from each other geographically.
- **Brainstorming**: this technique is used when the needs or requirements are not clear at the beginning, so that we can gather a few basic ones.

Let's see some of these techniques more in detail.

## 1.4.2.2. Interviews

An interview is a systematic way of getting information from another person, through a structured/guided conversation. This is an important aspect, since an interview does not consist in just asking questions. These questions must be ordered and structured, and they must follow a final purpose. The way we set the conversation will determine the information that we will finally get.

A good interview must be divided into these stages:

- **Preparation of the interview**, we get information about the customer company, identify our targets (people to interview) and prepare the contents of each interview, including date and time. Regarding the people chosen for the interviews, it is usual to start from the managers and go down to the employees.

We may need to include an additional "courtesy" interview with some section head, in order to smooth the way to interview his/her subordinates.

- **Interview**. This is the core of this process, and in this stage we distinguish three substages:
    - **Introduction**, where we introduce ourselves to the customer, we tell him about the main reason of the interview and how it will go. This way, the person interviewed will have a first impression about us and feel more comfortable from the beginning.
    - **Development** of the interview. At the beginning there may be more "open" questions, that lead to more specific ones (which can be answered with a simple "yes" or "no", or a simple short answer). We should adapt our speech to our listener (for instance, do not use technical words for someone who does not understand them), and repeat the answers given in order to verify them.
    - **Ending**, where we summarize the interview, thank our listener for his/her time, and leave an open door for a (possible) future second interview with him/her.

## 1.4.2.3. Joint Application Development (JAD)

This technique promotes the cooperation between the customer and the analyst. It consists of a set of meetings or workshops (2 to 4 days long each one), where some people from the customer company and the software company meet.

This way, we avoid fixing bugs or misinformations later, and we help the customer understand the process that the work team is following to develop the product.

Every workshop consists of the following stages:

- **Preparation**, where the project manager chooses the participants from both the customer company and the software company. He also gathers information about the system to be developed and organizes the meeting (place, date, materials needed...)
- The **JAD session** itself, which consists of multiple specific and structured meetings. At the end of this session a document with the requirements specification must be written, and it must be approved by every attendant.
- **Documentation**, from the requirements specification created in the JAD session, the attendants outline some additional details and provide a final, suitable format to the document.

However, this is not a very popular technique, because it requires higher organization, and breaking the customer's pace of work in order to attend the workshops periodically.

> **Proposed exercises:**

> **1.4.2.1.** Determine if the following statements are true or false, and why:
>
> - Interviews are the most suitable way of getting information for a software project, in general
> - JAD is a quick way of specifying the requirements of an application in projects where several departments of the customer company are involved
> - Questionnaires are useful when we try to gather information from the customer and the components are far from each other geographically

- Only with the observation of the company or the study of its documentation we can get a SRS very accurate.

# 1.4.3. Requirements specification

From the customer needs gathered through any of the techniques explained above, we must get to define the requirements of the application, either hardware or software. This way, we elaborate a document called **system requirements specification (SRS)**, which describes as much as possible what the software product is going to do.

Before going on, we must set what a requirement is: a **requirement** is a condition that the customer needs in order to solve a problem or reach a target. This way, we define the system restrictions.

For instance, a broker that asks us to implement an investment application may be worried about the response time of the application, which is crucial in order to buy or sell the stocks optimally. This quickness would then be a requirement for the application.

## 1.4.3.1. Requirement types

There are three main types of requirements:

- **System requirements**: physical components (hardware) which are needed in order to install the application. For instance, the number of computers to install the application in, number of processors/cores, hard drive space...
- **Functional requirements**: they set what the system is expected to do, including the inputs and outpus, user interactions, stored data... For instance, if the application must ask the user to log in at the beginning, or if it must send e-mails periodically to someone, they are both examples of functional requirements.
- **Non functional requirements**: they determine how well the system meets the functional requirements. In this category we can talk about performance, security... These requirements are less critical, since they don't make the application crash, they just degrade its performance.
    - In this type of requirements, a particular sub-category is the one related to the ease of use of the application, which measures the coupling between the application and the final user. In order to reach a good ease of use, we must analyze those final users, their level of skill, work environment... and also check the usability of the application screens (for instance, don't add too many buttons in a single screen), and some particular aspects, such as the usability for left-handed people.

**Proposed exercises:**

**1.4.3.1.** Read the following text about the development of a software product, and then identify the system, functional and non functional requirements:

*A blog has three types of users: administrators, editors and visitors. Any of them must log in in order to enter the application. Administrators can register other users, editors can publish posts, and visitors can comment them. We expect that this web site has lots of visits per day, so the availability of the service must*

*be high. We must also have several servers in order to balance the load and distribute client accesses. Besides, as we don't have very experienced editors, the user interface for them must be as simple as possible. Every post and comment will be stored in a MySQL database, with daily backups. Passwords will be encrypted in order to prevent possible attacks.*

**1.4.3.2.** From the following proposal for a software product, try to identify the system, functional and non functional requirements.

*A cultural organization is focused on the loan of two type of objects: music discs and books. We need an application that lets us add new objects to the system from both types. Besides, there are many users that come to this organization. They will need to log in with a user name and password. Then, they may be able to search any disc or book, either by object type (disk/book) and/or by object name (title). Once the object is chosen, they can ask for it if it is not available. Users can have up to 5 objects on loan simultaneously. As this organization is settled in a small town, we don't expect to have many users asking for books or disks at the same time, so we just need a single computer and a small database to store the information. In case a user does not remember his password, the system will send it by email to the same account that he entered when he registered.*

**1.4.3.3.** Classify the following requirements into system requirements, functional requirements or non functional requirements:

- The application must save the data before closing
- The database server must be in a separate computer
- If there is no connection with the remote server, the application must show an error message and then close
- The response time must not exceed 10 milliseconds
- We must have enough bandwidth to send the video in real time streaming
- We must verify that the information entered by the user (name, id and phone number) is correct