

Week 3. COM1030. Requirements Elicitation techniques.

Aims of the lecture:

1. Introduce the issue of a systems requirements.
2. Discuss problems in establishing requirements of a system.
3. Consider some practical methods of doing this.
4. Relate the material to the Crossover Project.

Most of all the key to requirements is:

good communication - between clients and developers, within the clients and within development team

analysis - what is the client wanting to do

inspiration - creating potential solutions to the client's problem

detail - is every aspect covered and clearly documented

practical - is it realistic, cost effective, timely

- above all DOES IT ADD VALUE?

The requirements for a system are a description of:

what the system must do

and how well it must do it.

If we get this wrong then the system we build will also be wrong.

It is very difficult to get right for the following reasons:

1. The client may not really know what he/she actually wants.
2. The developers may not understand the application/domain.
3. The client's business needs may change.

1. Researching the business background *Investigating the company background and bringing together relevant information.*

We will use Mindmaps - they can be used at many different levels to organise your thinking.

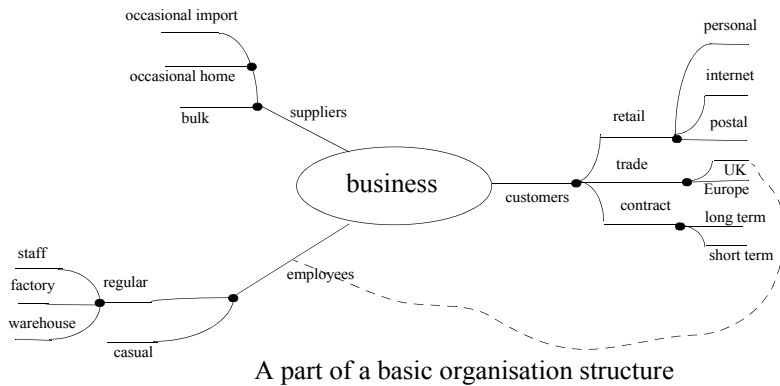
Mind Maps

A technique for noting and structuring information.

For these purposes business means organisation - so it will apply to the public sector and charities as well as commercial concerns.

We try to understand what the business is, what it does, its markets, customers, employees, suppliers etc.

Here is a sample analysis for a typical organisation drawn using mindmaps.



2. Exploring the outline system description

The initial project description may be nothing more than a paragraph and it might seem to be too vague to allow you to start.

Remember, however, that this description is just a starting point to you exploring, with the client, the client's business, its needs and possible solutions.

There will be a lot of preliminary work to do to define the scope of the project and what a potential solution might look like.

This is not an easy stage of any project and it is impossible to learn exactly how to do it in books and lectures.

The idea is to break down concepts over a number of stages.

Sometimes we might wish to identify a cross link - such as some employees are also customers purchasing at trade rates.

Other things to do:

Research into your client's business¹:

have they got a web page?

can you find any other published information about their business?

what do they sell - products or services or what?

what sort of clients do they typically have?

who are their competitors, what can you find out about them?

¹ In the Crossover project the client is your tutor so these steps are not really appropriate.

There is no substitute for trying it out and reflecting, as you go, on how the process proceeds.

Start with a brainstorm about the initial project brief and use *Textual Analysis* as well as mind maps, again.

LOONY TONES CORP

The company sells ringtones and video clips for use on the latest generation phones and personal devices through the web.

The company needs a regular supply of new products which are gathered from multiple sources.

A major concern is that some products are very similar to other products - the issues of copyright and illegal downloading is of major concern.

A system is needed that will check whether a ringtone or video submitted by a supplier, composer etc. is distinctive and not a copy of existing material.

We need to look closely at the phrases in this statement -

are they clear or ambiguous?

what further information do we need?

LOONY TONES CORP

The company sells **ringtones** and **video clips** for use on the **latest generation phones and personal devices** through the web.

The company needs a **regular** supply of new products which are gathered from **multiple sources**.

A major concern is that some products are **very similar** to other products - the issues of **copyright and illegal downloading** is of major concern.

A **system** is needed that will **check** whether a ringtone or video submitted by a supplier, composer etc. is **distinctive** and not a **copy** of **existing** material.

All of the highlighted phrases need to be checked out.

What are the range of products and prices - how will these be described?

How often will the product range be changed and by whom?

What sort of trend data do they want to extract and when?

What customer information is needed?

What is this accounts package?

All of these questions will need to be answered in detail.

Another example.

SmoothCore is a company that sells smoothies. They want a system that will allow them to manage their products and sales efficiently. They sell a wide range of smoothies in various quantities and at prices that vary according to the type and size of the drink. The system should allow the company to easily change the products and prices and to permit trends and customer preferences to be identified. The system will also manage the customer base and be seamlessly integrated into the company's existing accounting system.

We need to make a number of things clearer.

The statement also gives us some leads into both the functionality of the system and the data it will use.

Look for *verbs* - these will tell us about some of the functions of the system:

manage, change, identified, integrated,...

Nouns might guide us towards defining data precisely:

drinks, customers, products, prices,...

Adjectives may tell us how well the function needs to be:

efficiently, easily, seamlessly

From these initial steps we will drill down into the business and its processes in order to identify what needs doing.

The goal is to develop a detailed **Requirements Document** that will act as part of a contract between the client and the developers.

You will be creating one of these over the next few weeks.

This document will describe the background to the project including any current relevant information.

It will list the functional requirements - *what the system has to do*

and the non-functional requirements - *how well it has to do it*

in the later case this will address efficiency and speed, robustness, ease of use etc.

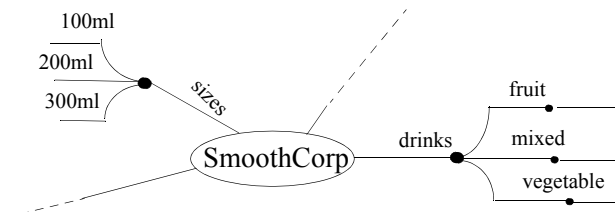
3. The first meetings with the client.

This might be a session involving all of the team working on that client's problem and generally involves the client describing their business and what they are trying to achieve, their objectives for the system.

There will be an opportunity to ask general questions relating to the system but these should not be technical computing questions - apart from general things such as the sort of network available or to be purchased for the solution.

Remember that the client may know very little about Computer Science or programming, that's why they have come to you, you are the experts. Their expertise is in their business.

We have seen some examples of outline project descriptions and *mindmaps* can also help to uncover more of the issues that need to be explored. Don't get into too much detail yet.



Starting with the initial project description there are a number of things you should do:

Preparing carefully for the first meeting will impress the client that you are professionals and will give them the confidence to proceed.

Initial stages of building a requirements document.

It is important that we aim to produce a clearly structured list of requirements, in language that the client can understand, so that an overall description of the complete target system is available.

This is developed in discussions with the client.

The key thing is to be aware that the requirements will change and to make sure that it is used as a summary of what the current knowledge of the proposed system is.

Requirements change is very hard to handle - new approaches to software engineering are being developed to address this.

The Agile methodologies such as Extreme Programming are claimed to be better able to handle change.

Kent Beck - Extreme Programming - embrace change.

Such approaches are much less bureaucratic than traditional software engineering. More of this next year.

4. Business analysis and problem discovery

The initial software requirements analysis can be divided into a number of activities:

Business analysis

Problem recognition;

Evaluation and synthesis;

Modelling and metaphor building;

Specification of user stories and scenarios;

Review and discussion.

Clients express problems naturally in their own words, words that might be unfamiliar to us or used in different ways, don't assume that your understanding of a particular word or term is the same as theirs.

We need to identify what the terminology means and to agree on it.

The construction of a *glossary* of business and technical terms should be an outcome of this dialogue.

When talking to clients realise that there may be hidden factors at stake: political, historical, geographical.

You may need to understand these features of a business organisation in order to understand the reasons for particular requirements.

In the first week or two of the project, you should be evaluating and synthesising the problem and requirements information from the client.

Always write down your thoughts, refer to these at your formal group meetings and put a date on them.

Later, you may need to revisit some issue when you have forgotten the details.

Although we wish to keep the paperwork to a minimum records of this stage should be saved, carefully.

Business analysis

What is the primary business objectives of the organisation?

How will a proposed system support these objectives?

Are there an existing systems - manual or computerised - that will be replaced?

Will the system interact with other systems - both internal and external?

What are the business processes and workflows?

How will the proposed system add value?

Systems will be involved that deal with:

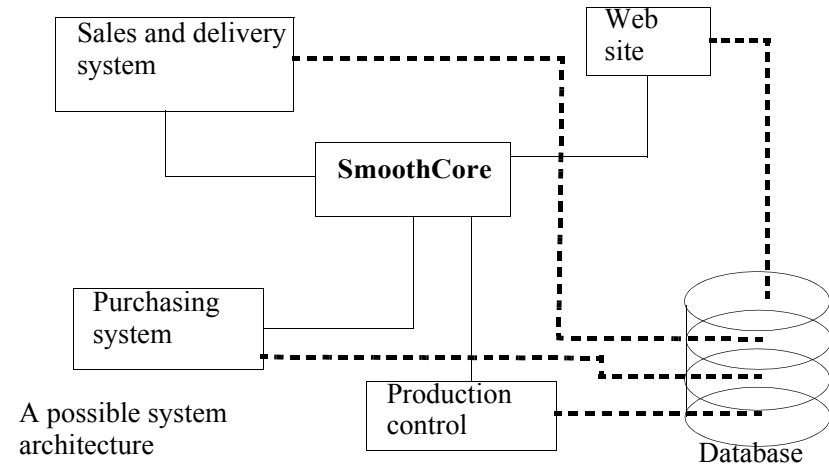
the purchase and payment of raw materials;
with customers orders;

with the factory production process

and a web site - that may include customer ordering capabilities.

Each system will have a database associated with it -

this may be implemented as a single unified database or as separate ones depending on the business needs.



Problem evaluation involves:

defining all external observable relevant business objects;

evaluating the flow and content of relevant information in the business;

defining and elaborating all relevant software functions;

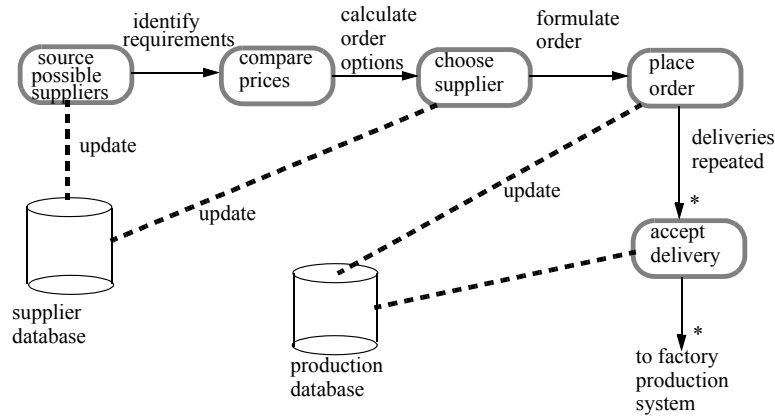
understanding relevant business behaviour (events);

understanding user behaviour (tasks);

establishing systems interface characteristics;

uncovering additional constraints.

Some process flows are indicated below.



Interviews have to be prepared carefully.

In the first meeting, when you know little about the problem, then it is important to ask the client to describe all the key aspects of the system.

Try to guide them away from the desire to get into intricate detail about what they want when you simply do not understand what they are talking about.

As you get immersed in their business context it is important to manage the meetings carefully.

Identify what you want to know beforehand and prepare a set of questions that will help you to find out what you need.

5. Techniques for requirements elicitation.

Here are 6 approaches that can be useful:

- Interviews;
- Structured questionnaires;
- Observation - again only successful, if you can do it unobtrusively;
- Concurrent protocols - a user describes his/her tasks whilst performing them;
- Card sorting - useful if you want to understand the user's classification of his/her knowledge domain;
- Carrying out a user role yourself;

Once these questions are answered then you can explore further areas.

It will often be the case that a question will stimulate the client into telling you some other piece of information, carefully record this.

It is best to go to the meeting with all the team but make sure that there is a principal speaker and someone to record what is said.

There is nothing more off putting for a client than to be faced with people asking questions from all angles on all sorts of disconnected topics.

Plan your meeting carefully and try to stick to it.

Structured questionnaire.

Using a written questionnaire is another technique.

Try to group related questions together. Also try to make your questions clear, unambiguous and relevant.

Observations.

Sometimes it is possible to visit the client and observe the business in action.

Here you may be able to observe users in their current work.

6. Putting your knowledge together.

Gathering all this information is one thing but putting it all together into a coherent model of the business is quite another.

There are no simple solutions to this problem.

Common sense is the best approach!

a) Defining all external observable relevant business objects.

We need to look at the sorts of things that are *coherent* entities in the part of the business we are considering.

This is helpful in providing you with a context and a better idea of what the users are like, what they expect or are comfortable with and what sort of system you might be trying to emulate.

Pay particular attention to the sort of user interfaces that seem popular.

Take care not to disrupt their work too much.

Some users are happy to talk their way through their tasks while you are there.

These could include: products, contracts, orders, invoices and such like.

Make a proper list of them and try to distinguish between those that are involved with the external activities of the business,

For example objects that are apparent to the customers, agents and suppliers of the business,

and to those involved in the monitoring of the company such as taxation and other government authorities

and the objects that are defined for the convenience of the internal management of the company, these might be: internal orders, memos and planning material, records and archives of company activity etc.

Many of these objects will relate to aspects of the databases that will be used to support the system activity.

b) Evaluating the flow and content of relevant information in the business.

Each business process will involve a number of individual processes which take place in an organised way.

What is the order in which this information is processed, what type of information is it?

Try to get a general picture of what happens and when during typical scenarios of business activity.

You will refer to the business objects described above, if you come across one that has not been identified, previously, then it needs to be added to the list.

c) Defining and elaborating all relevant software functions.

Now we can start imagining what our software is going to do.

It might be replacing some existing function, either a manual operation or in some obsolete software,

or it might be a new feature that has not been implemented with software before.

Equally, if you found an object that doesn't seem to feature in any process that you are analysing, eliminate it.

It may be that you find some difficulty in modelling things at the right level, there is always the temptation to try to describe things in too much detail.

Try to avoid this at this stage. We are looking for a rather "broad brush" description of what is going on.

d) Understanding relevant business behaviour (events).

Now we have to try to figure out how these things actually relate to each other.

We should try to define some common scenarios which explain the overall operation of the business processes through the medium of identifying the events that cause the scenario to operate.

These could be the placing of an order by a customer,

here we might need to identify what sort of customer is involved, a new or existing one, trade or retail.

The business process involved for each of these may be different and so the system will be expected to behave differently as well.

This leads to us identifying the different conditions that must apply for the different cases.

Again we need to check that our business objects and processes described above are consistent with this.

Alternatively, it can help identify aspects of the task which are currently problematic and could be improved in the required system.

As if requirements capture and analysis were not sufficiently complicated, we must often obtain the views of different users, who are likely to have different stakes in the outcome of the new system.

Hence, you may need to identify and resolve stakeholder views.

You should ask yourselves, who are your users?

They are not necessarily a single, homogeneous group of people with the same tasks, the same goals or the same view of the world. as the client commissioning the system.

e) Understanding user behaviour with task analysis.

There are many techniques for task analysis which can be used to elicit user requirements relatively easily.

Task analysis tends to concentrate on the way users conduct business processes now.

It may include user actions which do not involve interaction with a computer.

Nevertheless, a task analysis model can form a useful representation for discussion with your users, helping to identify aspects of the task with which users are comfortable and familiar with.

At the end of this process you should have identified the dependencies that the solution needs to relate to within the business context as well as any basic assumptions that pertain.

You may also have started to think about the constraints that will affect your solution, the available resources you have at your disposal, time, technology and so on.

This needs to be clarified, it is no use trying to specify a system that you are not able to build or the client cannot afford or operate.

A collection of requirements notes can be produced which can help to organise ones' thoughts into a more structured form.

The aim is to produce a detailed list of requirements which provides a basis for early planning and approval.

We will look at the initial descriptions of real projects that were recently done by DCS 2nd. year students

A full, detailed requirements document will be made available.

First we look at the clients' initial brief.

The Project

To produce a web-based system designed to support the FizziLink network of physiotherapists which also incorporates an automated patient registration and appointment booking system.

FizziLink

(the network of independent Chartered Physiotherapists)

Patient Registration & Appointment Booking System

Presented by:
Jim Mersh & Mike Self

Background

- ▶ 1000's of Physios working from small clinics throughout the UK
- ▶ Often difficult for patients to make enquiries or book appointments
- ▶ Physios responsible for their own administration e.g. letters, invoicing, accounts
- ▶ Need for regular communication and formal reports to GP's and health insurance companies
- ▶ Opportunities for economies of scale in other areas of business.

Web - Site Content

likely to include:

- ▶ Patient Registration Appointment booking facility
- ▶ Information Section including symptom analysis, exercise routines and preventative treatment
- ▶ Physio/Practice search facility
- ▶ Links to: BUPA, NHS Direct, Suppliers etc

Page 45

The requirement document will have a specific structure.

This is based on an international standard (IEEE).

Stage 1 of the Crossover Project involves you writing such a document.

You will be given an outline problem description like the ones above.

Your tutor will play the role of the client.

You will meet the 'client' and elicit the system requirements from them

Page 47

System Features

likely to include:

- ▶ Patient database
- ▶ Clinic & Physio database
- ▶ Diary for each clinic/Physio
- ▶ Appointment/Treatment Record
- ▶ Online Shopping
- ▶ Credit card Processing
- ▶ Reporting facilities

Page 46

We also classify each requirement as being:

mandatory - it must be present in the final solution;

desirable - it should be present if at all possible;

optional - only implemented if all others are done and there is still time.

Naturally the client will specify these levels and they may change during the course of the project.

Page 48

