

# Requirements Trawling: techniques for discovering requirements

Suzanne Robertson  
The Atlantic Systems Guild  
suzanne@systemsguild.com

## Overview

If you approach someone in the street and ask for directions then, providing that person knows the way – and speaks the same language as you do, it should be easy for him to help you. But often, when you try and follow the directions you become more confused and lost. Perhaps the person giving the directions has assumed that you know about a local landmark or has forgotten to mention that there is another small street on the left before the one you are seeking. Or maybe the director has not understood your request and has sent you to a place with a similar name....there are so many reasons that the transfer of information from one person to another is fraught with difficulties.

When you try to discover the requirements for any kind of product the difficulties are even more complex because the source of the requirements is not just one person, it is all of the people who are stakeholders in the project. And all of these people have their own view of what is important, along with their own experience, prejudices and views of the world. Considering the variations between your sources of requirements (stakeholders) it makes sense to have a variety of techniques for discovering the requirements. We call these trawling techniques because, like fishing, we run a net through the organisation and trap as many requirements as we can. Then, using the appropriate technique we identify the relevant requirements (the juicy codfish) and separate them from the irrelevant (the minnows). We also look for rare and amazing fish that nobody has ever seen before. We are not just concerned with finding existing requirements we are also concerned with generating new requirements by using techniques that encourage creativity. This paper summarises a number of techniques that we have found useful when trawling for requirements.

## The problem of requirements - Why didn't you tell me what you want?

A requirement is some capability that somebody needs or wants. This is rather a “catch all” definition because there are many different types of requirements and there are also goals and constraints which, it is useful to think of as special types of requirements. The aim is to discover all the requirements that are relevant to the product that we intend to build as early as we possibly can. However it often happens that requirements which could have been known about earlier are not discovered until after the product has been built. The reason for the late discovery of requirements is usually because we have not been able to inspire the stakeholders to think past preconceptions and communicate what they

The type of requirement that a stakeholder is most likely to communicate is what we call a *conscious* requirement. A conscious requirement is something the stakeholder is particularly aware of. This might be because it is one of the reasons for building the product like “I want the camera to fit in my handbag”. Or it might be because a current product has a shortcoming – “I want the battery to last longer”. Or maybe it is because the stakeholder is aware of a new piece of technology – “I want the camera to be able to take digital photographs”. In all these cases a particular stakeholder is conscious of a requirement because of his particular view of the world, consequently it is something he is likely to mention.

Another situation is when a stakeholder does not mention a requirement because he does not realise that he has it. Think of it as an *unconscious* requirement. Reasons for this

situation might be that the stakeholder is so used to having this requirement satisfied that he no longer thinks of it as a requirement. If he is so used to his camera automatically rewinding at the end of the film, then he is less likely to articulate that requirement. But if you deliver a camera that does not rewind then he is amazed. How could you possibly have missed that requirement when surely anyone would know it is necessary. The problem of unconscious requirements happens often when the stakeholder knows a lot about the subject matter and makes assumptions that everyone else has the same knowledge. Another reason is because it is sometimes very difficult to tell someone all the details about something you know a lot about because you feel that maybe you are patronising them. You feel that surely they know that and you might be boring them by even mentioning it. Yet another reason might be that the stakeholder does not understand what an existing product does and therefore assumes that any new product will maintain the status quo.

*Undreamed* of requirements are rather different. If a stakeholder has a fixed idea of what he believes is possible then he is unlikely to mention requirements that he thinks cannot be carried out within his understanding of the constraints. “No point in mentioning that I would like the camera to be waterproof – I know it’s not possible within the budget”. Then there are many undreamed of requirements that do not even occur to the stakeholders because they cannot imagine what it might be like to have the product and to experience new technology. Remember that many of these undreamed of requirements will not be invented by stakeholders who fall into the category of customers or end users. Instead other stakeholders like technical specialists and developers will be the inventors and suggestors of undreamed of requirements. Unless we encourage stakeholders to imagine these undreamed of requirements, they are unlikely to surface until later in the product’s lifecycle when people understand more about the potential uses of new technologies. By then, even though it might have been possible earlier, it is often too late to add these new competitive features to the product.

## **A variety of techniques for discovering requirements**

The most traditional technique for trawling for requirements is interviewing the stakeholders. The intention is that you ask the stakeholders what they want and they tell you their requirements. Although this is a useful technique in many situations, for all the reasons already discussed this one technique cannot hope to uncover all the requirements.

The number of trawling techniques we use is growing and will continue to grow as we become more ambitious in terms of the size, complexity and level of human involvement of the products that we build. We need ways of dealing with the fact that requirements come from more and more different people with different experiences and concerns in different parts of the world. For many of the products that we build it is impossible to talk to all the stakeholders hence we need a variety of other techniques for discovering their interests. Figure 1 identifies techniques that we have found useful, I will summarise them, give some guidance on when a particular technique is most appropriate and provide some further references. For detailed surveys of other trawling techniques that I do not mention due to space limitations I refer you to [Maguire ’98, Maiden & Rugg ’96]

### **A Variety of Trawling Techniques**

- Â Abstraction
- Â Apprenticing
- Â Business Events
- Â Brainstorming
- Â Family Therapy
- Â Mind Mapping
- Â Interviewing
- Â Neurolinguistic Programming
- Â Reusing Requirements

- Â Simulation Models (scenarios, prototypes)
- Â Soft Systems
- Â Systems Archaeology
- Â Use Case Workshops
- Â Video
- Â Viewpoints

*Figure 1. Trawling techniques to help you uncover conscious, unconscious and undreamed of requirements*

### **Abstraction**

The technique of abstraction is based on making useful generalisations. The purpose of the generalisations is to understand and articulate the rules that apply to a specific domain of knowledge and even to discover rules that are shared between domains. Many people find it very difficult to think in terms of abstractions and are much more comfortable thinking in terms of specific instances. For instance, suppose someone says “If I press this button on the top of my camera then it takes a photo”. A more abstract view might be “all devices that take photographs have some way to set in motion the taking of a photograph”.

People who are good at abstract thinking are usually comfortable looking at class models, data models and other models that focus on the “essence” [McMenamin & Palmer '84, Robertson & Robertson '94] of the subject matter.

### **Apprenticing**

Apprenticing uses the idea of a master craftsman and an apprentice. The apprentice observes what the master does, asks questions and then tries to learn the work by doing some of it. [Beyer & Holtzblatt '98]

Apprenticing is a good technique when stakeholders say they do not have enough time to be involved in the project. Instead the requirements analyst involves himself in the stakeholders' work by becoming part of it. The technique also works very well when the stakeholder is having trouble articulating his goals or when his knowledge is limited to a small part of the system.

### **Business Events**

The specification of requirements always involves the investigation of some activity, work or business in the world. Size and complexity means that it is too difficult to investigate all of the details of this work as one large task. Instead you can identify the boundaries of the work, partition it into a number of logically related chunks and then investigate the details of each chunk. We use the idea of business events” [McMenamin & Palmer '84, Robertson & Robertson '94] as a way of discovering, investigating and managing these logically related chunks or business event responses. This technique has some similarities to task analysis and decomposition methods and results in a functional partitioning of a complex whole.

### **Brainstorming**

The purpose of brainstorming is to use the group effect to generate good ideas and to solve problems. The technique focuses on rapidly generating as many ideas a possible without stopping to evaluate or clarify. Many of the ideas will be wild, crazy or impractical, but that does not matter because they provide the trigger for discovering and inventing undreamed of requirements. After the brainstorm evaluate the ideas you have come up with and choose the ones that best suit your stakeholders. Guidelines for brainstorming [Bolton '79] are:

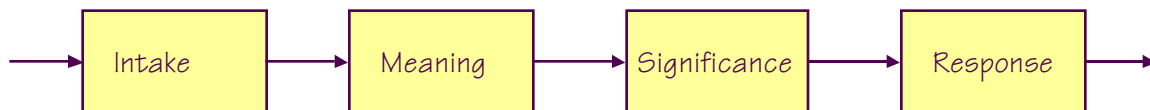
1. Don't evaluate
2. Don't clarify or seek clarification
3. Go for zany ideas

4. Expand on each other's ideas
5. List every idea
6. Avoid attaching individual names to ideas

Brainstorming is a particularly effective technique if you are inventing a product and you are trying to explore and understand the potential market.

### **Family Therapy**

The stakeholders for a project are bound together rather like the members of a family. There is a commonality because everyone is in the same social structure – in our case the project. However individual members have very different ideas about what is important and what they should contribute. Family therapists [Satir '91 and Hoffman '81] have been working for many years to understand and help family groups to achieve a peaceful and productive co-existence.



*Figure 2. Virginia Satir's Interaction Model*

Figure 2 is an example of a family therapy model that requirements engineers can use to help them communicate with other people. The model illustrates the 4 basic parts of any communication:

1. Intake – we make a choice about what to hear or observe
2. Meaning – we assign our meaning to the data
3. Significance – we decide how we feel about the meaning that we have assigned.
4. Response – we make our chosen response.

The field of family therapy is a rich source of ideas about how to work effectively with a diverse group of people such as stakeholders in a requirements trawling process. We use family therapy as a way of helping us to listen to stakeholders, provide a feedback loop to help avoid misinterpretations.

### **Mind Mapping**

When you are observing, listening or interviewing you normally take some kind of notes to help you to recall and refine your understanding. However taking notes that you can make sense of afterwards is not easy. It's also difficult to include all the details when you are writing in everyday language and trying to keep track of what is said. The mind map [Buzan '93] is a way of taking more extensive and meaningful notes. With a mind map you use words, pictures, symbols and colour to capture the way that your brain perceives the subject matter. Later, when you look at this personalised view you will find it much easier to recall details that you would miss if taking notes using a standard linear language style.

We have found that mind maps are a wonderful way of enriching the notes we take during interviews. They also work well as a tool for summarising your understanding of something that you have read or for taking notes during a lecture.

### **Interviewing**

An interview is probably the most common way of gathering requirements. This technique can be very effective but is often misused. The best way to use interviewing is when you have the opportunity to talk to an expert about a bounded area of subject matter. In other

words it is better to go to the interview with a well-formed intention of what you wish to achieve.

Some guidelines to follow are:

1. Define the purpose and contextual boundary of your interview so that you can keep it on track
2. Have a fixed time limit
3. Talk to the people who have hands on experience with the subject
4. Use models and sketches to feedback and improve your understanding
5. Listen to your interviewee
6. Illustrate the relevance of the interview by highlighting something significant that you have learnt
7. Thank the interviewee for giving you the time

To improve your interviewing skills, try focusing on becoming better at having relevant conversations. One resource is the work of the Dialogue Group [Ell '98] that is devoted to improving conversational and collaborative skills.

### **Neurolinguistic Programming**

Neurolinguistic programming [O'Connor & Seymour '88] is a set of models, skills and techniques for thinking and acting effectively. The technique has many different elements including influencing skills, understanding body language, the art of asking key questions, effective meetings and negotiations. One of the elements of neurolinguistic programming is a meta-model for getting a better understanding of what people say. The model is made up of a number of patterns together with clarification questions to help establish meaning.

For example, one of the patterns is called "unspecified noun". You find this pattern when you come across a noun that does not specify an exact instance and hence is open to interpretation. If you hear, "*the place was pleasant*", you do not know precisely which place the speaker is talking about. In this case the noun "place" can be considered an unspecified noun. The clarification question for the unspecified nouns pattern: Who or what specifically...? In this case, what place are we talking about? This line of questioning unpacks the meaning behind "*the place was pleasant*" and leads to a more specific statement like "*the park was sunny and full of tulips*".

You can use the neurolinguistic meta-model as a tool for identifying vagueness or ambiguity in your requirements.

### **Reusing Requirements**

If you define your requirements using a consistent approach, then you have the opportunity to reuse requirements. You can discover reusable requirements at many different levels of detail.

Suppose, for example, you have an atomic requirement with a well-defined fit criterion (measurement). Then it's possible that the same requirement could be relevant either in another part of the same project or in a different project. Other examples of reusable requirements might be the definition of a class of data or the detailed definition of one business term. If you start looking for recurring patterns of requirements you often find an opportunity to reuse knowledge that has already been defined. When you define a business event-response or a product use case you have defined a number of requirements that are connected by a common purpose. For example if you defined a number of requirements that relate to a customer wanting to open an account in a bank, then it's likely that some or all of those requirements could be reused in another similar situation like opening an account with a department store or an Internet service provider. Refer to [Robertson & Robertson '94 and Hay '95] for more details about requirements reuse. Also read [Jackson 01] for ways of reusing problem frames.

## **Simulation Models– scenarios, prototypes**

When used as a requirements trawling technique, the intention of a simulation is to tell a story [Schank '90] to make something appear to be real for the purpose of stimulating requirements or ideas that might otherwise be forgotten or overlooked. The most effective simulations are designed according to the characteristics and knowledge of the intended audience. You aim to include details that are relevant to those peoples' view of the world to help them view the simulation as if it is real. If you can pull off this conjuring trick then you will generate many requirements that would otherwise not be mentioned until the real product is installed.

A successful simulation model tells a story that can either focus on the normal course of events (the abstract or general view) or a specific case that includes values for all the data. Techniques for building simulation models include sketches, storyboards, prototypes, and scenario models. For more on the technique of paper prototyping refer to [Nielsen '90].

## **Soft Systems**

Soft systems [Checkland '81] is a systems thinking approach for observing and modelling the world for the purpose of understanding and tackling real-world problems. The acronym, CATWOE, provides a checklist for the aspects of the model. These are:

- Customer – the beneficiaries or victims of the system
- Actors – those who carry out the transformations within the system
- Transformation – of some defined input to defined output
- Weltanschauung – the image of the world that makes this system meaningful
- Owned – the owner/s of the system
- Environment – the environmental constraints

Systems thinking techniques are sorely needed by requirements engineers as a way of helping to take a wider view of the world and thereby discover or create requirements earlier in the lifecycle of a product.

## **Systems Archaeology**

One source of requirements is documents that relate to an existing business system or piece of software. Systems archaeology is a technique for extracting requirements from existing documents. Imagine that the documents are clues about a past civilisation, you can't talk to the people but you can imagine what their lives might have been like by digging through the layers of artefacts that they produced. Ideally, in order to get started in understanding the terminology, someone would introduce the document to the requirements engineer.

A data model [Shlaer & Mellor '88] is a useful tool for recording the results of your archaeological study and raising questions about the underlying requirements.

## **Use Case Workshops**

A use case workshop provides a focused way of reviewing and questioning a related group of requirements and coming up with ideas for part of the product. The starting point for such a workshop is a bounded chunk of functionality that can be identified as a business event response or a *business use case*. For example a library project might have a business use case called *add book to catalogue*. From the business use case's point of view this includes all the business related knowledge necessary to carry out this work along with the characteristics of all the stakeholders involved. During the workshop stakeholders who have the knowledge about this business use case review the business requirements and come up with new ideas for how the business could/should respond to this business event. This leads to discussion of ideas for the product.

The *product use case* is that part of the business use case that you decide will be within the boundary of the product. You negotiate the product use case by considering the goals of the project, the constraints and the preferences of the stakeholders for this use case.

## **Video**

We have used video in usability laboratories to record users' reactions to software products. The idea is to derive new requirements, especially those related to usability, by evaluating individuals' facial expressions, words and body language while they are using the software. For example we have seen video used as a trawling technique with air traffic controllers. The technology is successful in this environment because the stakeholders are accustomed to it. As people in other environments become more used to video, requirements engineers can make wider use of this technology.

For example, we could set up a video camera in a stakeholder's workplace then we could review the results for the purpose of understanding more about the work and generating relevant questions. Another idea is to make a video of a use case workshop and use it to familiarise people with that subject matter. There are many other ways that we could take advantage of video but, like any new technology, it takes a while for people to be comfortable with the idea. One vital ingredient is to have a formal agreement stating that the video will only be used for purposes authorised by the people who appear in it.

A step towards video is the use of still photography. We often, with the stakeholders' consent, take photographs of a working environment. We explain that we want to study the photographs so that we can think of relevant questions to ask at the next meeting. We also say that we can use the photographs to familiarise other people in the project team with the working environment.

## **Viewpoints**

The complexity of requirements necessitates a way of independently focusing on different points of view. If you can do this then you have more chance of discovering requirements. However, in order to be able to trace requirements throughout the product's lifetime, you also need to be able to make connections between the different points of view.

Some viewpoints [Robertson & Robertson '94 & '99] [Jackson '96] that are particularly useful are the view of the world that you are trying to understand, the view of the world as it will be in the future and the view of the product. The first two views include all aspects of the world that you need to understand in order to build a relevant and competitive product. The view of the product or the machine, as Jackson describes it, focuses on the product boundaries and its intended connections to users and other products. It is much easier to discover requirements if you are able to focus on different views of the world rather than trying to think of everything simultaneously.

## **What are we Looking for?**

It is useful to consider which requirements-related components you need and how they should be related to each other so that you can keep track of project progress.

*Functional* requirements are things that a system has to do, like record a fact, do a calculation or make a decision. Functional requirements exist because of the subject matter within the context of the particular system. An airport system would have a functional requirement to record the landing time of a flight, a gardening system would have a functional requirement to fertilise the plants, a bottling system would have a functional requirement to fill the bottles.

*Non-Functional* requirements (sometimes called qualities or attributes) are the qualities that a system has to have. Things like performance, security, usability, maintainability are all non-functional requirements. How fast does the bottling system have to fill a bottle, how convenient must it be for the gardener to fertilise the plants, who is permitted to look at the information about flights...these are all non-functional requirements.

*Project Goals* are the reasons for doing a project. You can think of these as a type of high level requirement – all the other requirements contribute to meeting the project goals. The gardening system might have a goal of increasing the annual harvest of vegetables by

50%, maybe the airport's target is to accommodate 20% more passengers and the bottling system aims to reduce the number of defective products by an agreed amount.

*Constraints* are specified influences that affect the way that we meet the requirements. Perhaps the new airport system has to be operational within 1 year, the gardening system must only use organic products and the bottling system has a budget of 1 million Euros. The most common constraints are time, money and specified technology. There are ongoing discussions about whether a constraint is or is not a requirement. It is more productive to take the approach that, in order to keep a project relevant and on track, it is necessary to quantify constraints and correlate them to the requirements. To make sure that a stated constraint really is a constraint we need to ask the question "why is it a constraint?". Why must the gardening system only use organic products? If the answer is because we thought it would be a good idea then it sounds more like a solution than a real constraint. However if the answer is that our environmental policy is to use only organic products then it sounds like a real constraint.

## **A Template for Guidance**

I have talked about the necessity of having a variety of trawling techniques and we can see how different techniques could produce requirements in a number of different forms. While you need this freedom in order to discover requirements, you also need some way of reviewing the requirements and making them consistent. If you have consistency, then you can base your project planning and monitoring decisions on your knowledge of the requirements. A requirements specification template is a useful guide for helping your project to look for, organise, and review the requirements. The following is the table of contents of the Volere requirements template. The template (see Figure 3) is a guide based on our experience with requirements specifications on many different types of project in a wide variety of industries. The content, currently 51 pages, contains the generic contents of a requirements specification along with examples and discussion of different types of requirements and constraints, how they relate to each other and how requirements knowledge can be used as input to project management.

We published the first version of the template three years ago, since then we have had feedback from many different organisations all over the world. Each new version of the template is updated with the experiences from many projects. To give you an idea of the diversity, some examples of projects using the template are to specify requirements for: web pages for a multimedia publishing company, an organisation installing SAP, air traffic control systems, a large multi-national manufacturing system, defence systems, replacement of a legacy banking system, new releases of system software, a building – this was by civil engineers and had nothing to do with software. Universities in Europe, U.S.A. and Australia have adopted the template for use in their requirements classes

You can download the template from our web page <http://www.systemsguild.com>

# Volere Requirements Specification Template

## **PROJECT DRIVERS**

1. The Purpose of the Product
2. Client, Customer and other Stakeholders
3. Users of the Product

## **PROJECT CONSTRAINTS**

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



## FUNCTIONAL REQUIREMENTS

7. The Scope of the Work
8. The Scope of the Product
9. Functional and Data Requirements

## NON-FUNCTIONAL REQUIREMENTS

10. Look and Feel Requirements
11. Usability Requirements
12. Performance Requirements
13. Operational Requirements
14. Maintainability and Portability 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. Cutover
23. Risks
24. Costs
25. User Documentation and Training
26. Waiting Room
27. Ideas for Solutions

*Figure 3. The Volere Template*

<http://www.systemsguild.com/GuildSite/Robs/Template.html>

Items 1 – 17 contain different types of requirements-related knowledge. Items 18 to 27 contain project-related knowledge that is derived from an understanding of the requirements.

### **Organising the Requirements**

The template helps you to discover the requirements but you also need to keep track of how the requirements relate to each other. To use the requirements as a project planning and monitoring tool, you need to be able to measure how many of each type of requirement you already have and how many more you are expecting to find. These sorts of measures are best done using some kind of automation - there are many different tools on the market [see [www.systemsguild.com](http://www.systemsguild.com) for a summary]. But before you rush out and buy a tool it is best to have a clear idea of what you are trying to measure and why.

Figure 4 is a model of the types of requirements knowledge mentioned in the template, together with the connections between them. Each box represents a class of knowledge about requirements; the numbers refer to the section numbers on the requirements template; the lines between the boxes indicate a relationship between the types of knowledge; the annotations indicate why the relationship is useful. For example there is an *Implementing* relationship between *Business Event* and *Product Use Case* so that we can see which parts of the product implement requirements for which parts of the business. Similarly there is a *Detailed Product Tracing* relationship between a *Product Use Case* and potentially many *Requirements* (and vice versa) so that we can assess how a change to any one of the requirements will affect the product.

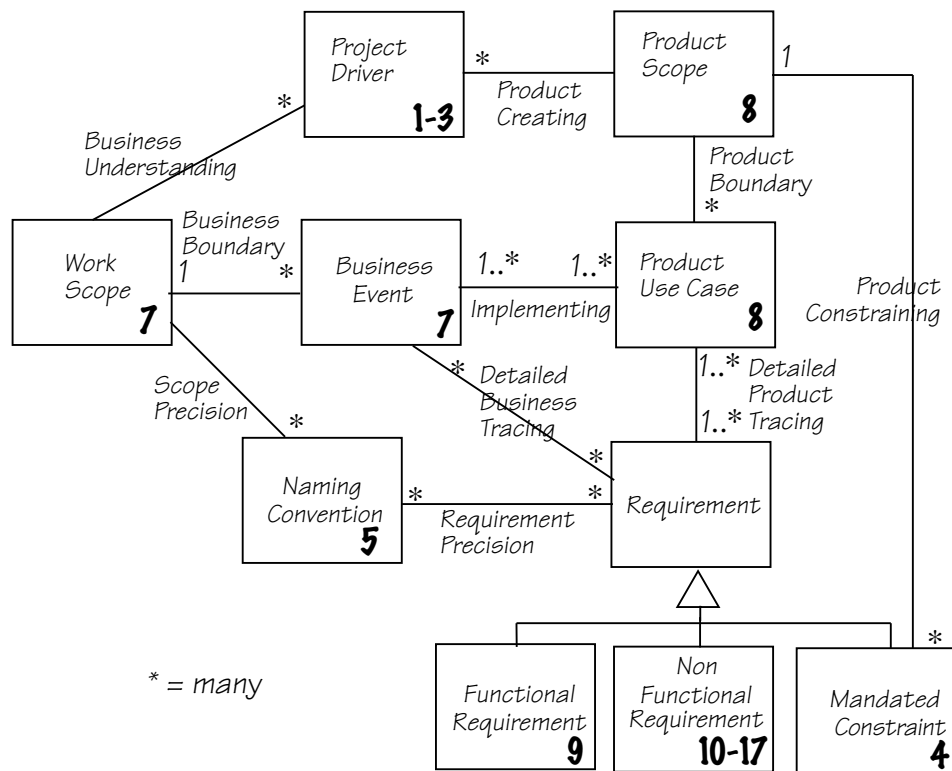


Figure 4: A model of the classes of requirements knowledge that you can use to plan and monitor the project.

### What's in it for Me?

If you gather and organise your requirements so that you can precisely identify the types of knowledge and the relationships between them then you can monitor the completeness and consistency of the requirements and use that knowledge to make and tune your project plan.

You can assess how much you already know about the requirements and where the most difficult problems might be lurking, if you start the project by attempting (along with the guidance in the template) to specify items 1 to 8 on the template

1. The Purpose of the Product
2. Client, Customer and other Stakeholders
3. Users of the Product
4. Mandated Constraints
5. Naming Conventions and Definitions
6. Relevant Facts and Assumptions
7. The Scope of the Work (Business Events)
8. The Scope of the Product (Product Use Cases)

This provides you with a first cut assessment of how much you know about the requirements and identifies the gaps in your knowledge and where you need to go to learn more. You can use this knowledge to design tasks for your project team – Jane will investigate the first ten business events, Bill will look after the next ten and meanwhile Cassandra will talk to the stakeholders from the marketing department to gather their aims and ideas for the scope of the product.

Figure 5 shows the components of an individual functional or non-functional requirement. As your project progresses and you gather the detailed requirements then you

can assess the quality of individual requirements and test them for accuracy, completeness and understandability.

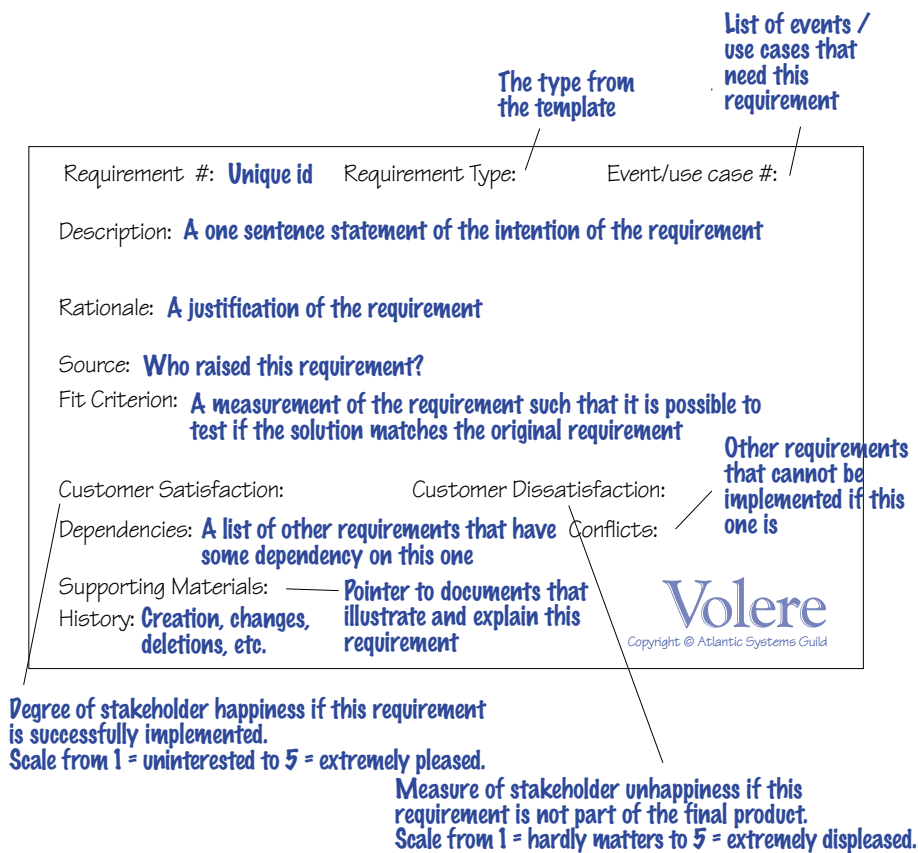


Figure 5: You can test individual requirements to ensure that you have all of the necessary components

Each product use case is a cluster of requirements. Providing you keep track of the relationships we have discussed, you can use these clusters of requirements as mechanisms for planning releases of your product. The first version will implement one identified set of product use cases; the next version will add the next ones on the list and so on.

Another advantage of this approach is that you can involve testers earlier in the project. Because you have an objective specification of exactly what you mean by a requirement, then testers will be able to test requirements to ensure that they really are requirements. They can also assess clusters of requirements to determine whether it is possible to build a cost-effective test to test the eventual solution to the requirements.

### Choosing the Appropriate Technique

I have already mentioned the number of trawling techniques we use needs to continue to develop to keep pace with our ambition in terms of size, complexity, fragmentation and level of human involvement in the products that we build. With the growth in the number of techniques available comes the question, which technique is the best one in which set of circumstances? There is no simple answer to this because the choice of technique is driven by the characteristics of the knowledge source, and in most cases that means the characteristics of individual people. This situation leads to the necessity of using the techniques in combination. To aid these choices, it is helpful to think of the strength of a technique in terms of its ability to discover conscious, unconscious and undreamed of requirements.

For instance apprenticing is likely to uncover unconscious requirements because the requirements engineer will observe requirements that nobody mentions because they are very familiar. Brainstorming is very good for undreamed of requirements because it helps to free people from preconceived ideas and encourages them to dream. When you interview someone they are most likely to tell you about their conscious requirements because those are the things that are uppermost in their minds. Figure 6 summarises the trawling techniques along with indications of what we have experienced as their relative strengths

Trawling Technique	Conscious	Unconscious	Undreamed
<i>Strategy and Planning</i>			
Business Events	3	2	1
Family Therapy	1	3	1
Soft Systems	2	2	2
Viewpoints	2	2	2
<i>Capture and Generation</i>			
Apprenticing	2	3	1
Brainstorming	1	1	3
Interviewing	3	2	1
Mind Mapping	2	2	3
Simulation Modelling	1	2	3
Systems Archaeology	1	3	1
Use Case Workshop	3	2	2
Video	2	2	2
<i>Analysis</i>			
Abstraction	1	2	2
Neurolinguistic Programming	1	3	1
Reusing Requirements	2	2	1

Figure 6: This table groups the trawling techniques according to whether they are most applicable to the activities of Strategy and Planning, Capture and Generation or Analysis. Then each technique is graded according to its relative strengths in discovering conscious, unconscious and undreamed of requirements. 1 = limited effectiveness, 2 = effective, 3 = very effective

## Conclusion

We need a variety of trawling techniques to help discover requirements from many different sources. And you need the freedom to be able to react to your own environment and to collect requirements in the order in which they occur. A defined requirements structure gives you the formality necessary for a common way of communicating about requirements. However the formal structure should not force you to work in a procedural manner, it merely provides pigeonholes so that you know where to put particular knowledge when you discover it. So you can always identify gaps and changes in the requirements specification and adjust your plan accordingly. And lastly, because you are writing the requirements in non-technical, albeit precise, language you can focus the involvement of the appropriate stakeholders in the parts of the specification that are directly relevant to them. Your project team can get more accurate answers because the stakeholders understand the requirements rather than having to wait until the product is built and then saying “Hmmm, what I really meant was.....”.

## References:

- Beyer, Hugh and Karen Holtzblatt. *Contextual Design. Defining Customer-Centered Systems*. Morgan Kaufmann, San Francisco, 1998.
- Bolton, Robert. *People Skills. How to Assert Yourself, Listen to Others and Resolve Conflicts*. Simon & Schuster, 1979.
- Checkland, Peter. *Systems Thinking, Systems Practice*. John Wiley & Sons, Chichester, 1981.
- Ellinor, Linda and Glenna Gerard. *Dialogue: Rediscovering the Transforming Power of Conversation*. John Wiley & Sons, New York, 1998.
- Gause, Donald and Gerald Weinberg. *Exploring Requirements: Quality Before Design*. Dorset House, New York, 1989.
- Hay, David. *Data Model Patterns: Conventions of Thought*. Dorset House, New York 1995.
- Hoffman, Lynn. *Foundations of Family Therapy*. Basic Books, U.S.A., 1981.
- Jackson, Michael. *Problem Frames: analyzing and structuring software development problems*. Addison Wesley, London, 2001.
- Jackson, Michael. *Software Requirements & Specifications: a Lexicon of Practice, Principles and Prejudices*. Addison-Wesley Longman, Wokingham, England, 1996.
- Kovitz, Benjamin. *Practical Software Requirements – a Manual of Content & Style*. Manning, Greenwich CT, 1999.
- Leffingwell, Dean and Don Widrig. *Managing Software Requirements - a Unified Approach*. Object Technology Series (Series editors Booch Jacobson Rumbaugh), Addison-Wesley, 2000
- McMenamin, Steve & John Palmer. *Essential Systems Analysis*. Yourdon Press, New York, 1984.
- Maguire, M. (1998). EC Telematics Applications Programme Project TE 2010 - *RESPECT user requirements framework handbook*. HUSAT Research Institute, Loughborough University, The Elms, Elms Grove, Loughborough, LE11 1RG, UK. An early version is available online at: <http://www.lboro.ac.uk/research/husat/respect/rp2.html>
- Maiden, Neil & Gordon Rugg. *ACRE: selecting methods for requirements acquisition*. Software Engineering Journal. May 1996.
- Nielsen, J. 1990, *Paper versus computer implementation as mock-up scenarios for heuristic evaluation*, INTERACT'90 conference, pp 315-320; Elsevier, North Holland
- O'Connor, Joseph & John Seymour. *Introducing Neuro-Linguistic Programming*. Thorsons, London, 1990.
- Robertson, James & Suzanne. *Complete Systems Analysis - the Workbook, the Textbook, the Answers*. Dorset House, New York, 1994.
- Robertson, James & Suzanne. *Mastering the Requirements Process*. Addison-Wesley, London, 1999.

Satir, Virginia et al. *The Satir Model, Family Therapy and Beyond*. Science and Behaviour books, Palo Alto, California, 1991.

Schank, Roger. *Tell Me a Story – Narrative and Intelligence*. Northwestern University Press, Evanston IL. 1990.

Shlaer, Sally and Steve Mellor. *Object-Oriented Systems Analysis*. Yourdon Press, Englewood Cliffs, New Jersey, 1988.

Weigers, Karl. *Software Requirements*. Microsoft Press, 1999.

For a variety of articles about requirements

<http://www.systemsguild.com>

To download the Volere template

<http://www.systemsguild.com/GuildSite/Robs/Template.html>

For other web sites and discussion groups

<http://www.systemsguild.com/GuildSite/Robs/ReqsEngResources.html>

For a list of requirements tools <http://www.systemsguild.com/GuildSite/Robs/retools.html>

## **Suzanne Robertson**

Suzanne is co-author, with James Robertson, of *Mastering the Requirements Process* (Addison-Wesley 1999) a book that provides guidance on finding requirements and writing them so that all the stakeholders can understand them.

Suzanne is a principal of The Atlantic Systems Guild, a systems think tank. She has more than 30 years experience in project management, system specification and building. Her courses on requirements, systems analysis, design and problem solving are well known for their innovative workshops and business games.

Current work includes research and consulting on patterns and the specification and reuse of requirements and techniques for assessing requirements specifications. The product of this research is *Volere*, a complete requirements process and template for assessing requirements quality, and for specifying requirements.

Suzanne has recently been appointed as the first editor of the new Requirements column for IEEE Software magazine.