Chris Rupp, Dirk Schüpferling

5

Eliciting Requirements – Clairvoyance for Connoisseurs

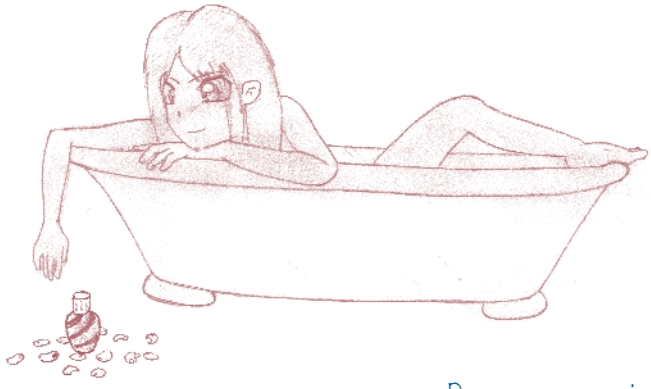
- He who does not know what he desires shall not be surprised by what he receives.
- Every elicitation technique has its pros and cons.
- Without a closer look at the characteristics peculiar to a project, no efficient use can be made of the elicitation techniques.
- Combining different elicitation techniques is the key to your project's success.

5.1 Cuddle those customer demands

Eliciting requirements is an act of seduction. You must seduce your stakeholder into sharing his knowledge, his visions with you. It's a bit like a flirt, because flirting is a game were you won't know if you're still in the qualifying round or already in the finals. Of course, this can also be expressed in more technical terms.

Of course that wouldn't be half as much fun...

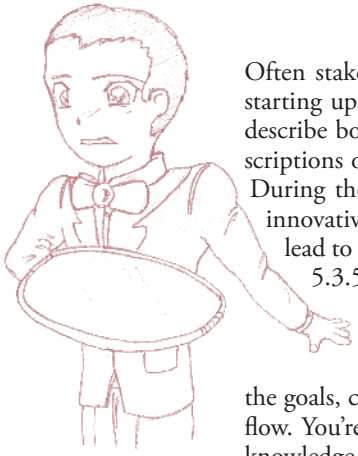
The overall objective is to elicit the goals and requirements with as little time and effort as possible while staying within the general framework of the project, in order to develop a system which will generate as much profit as possible for the stakeholders. For these reasons, we're looking for the golden path between risk reduction and cost explosion while on the lookout for the most professional way to make the seduction come true.



Requirements aren't handed to you on a silver platter.

5.1.1 The first steps are always the hardest

All requirements, independent of their level of detail, their type or the point in time they were elicited, must be obtained from a knowledgeable source.



Often stakeholders won't know what to expect from a new system when a project is just starting up. They do know the current business processes and the existing system, and can describe both (actual situation). What you need to do is abstract the usually pragmatic descriptions of solutions the stakeholders utter, in order to acquire the essential requirements. During the development phase, these essential requirements can then be used to develop innovative solutions and delineate new pragmatic processes. These will then (hopefully) lead to an optimized version of the processes (for more details please take a look at section 5.3.5 "Reduction to the essence").

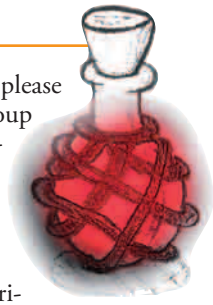
As a requirements engineer it's your job, jointly with the stakeholders, to define the goals, constraints and requirements of a system which will support an optimized process flow. You're the host who has to ensure the stakeholders´ time is efficiently used to gather knowledge. When you're hosting, make sure you're not also imparting ideas of your own –

don't mix these roles. You have to work as a catalyst which helps the stakeholders generate ideas and helps them realize what they really want. But don't expect the stakeholders to hand over perfect goals, constraints or requirements on a silver platter.

An old Chinese saying goes: It's not enough to come to the river wishing to fish, you need to bring a net.

As a requirements engineer it's your job to help stakeholders get a clear grasp of what they actually want. You must help unearth knowledge from the subconscious and the unconscious.

5.1.2 Spoilt for choice



In the course of a project the most diverse types of requirements (for more details please take a look at chapter 1 "In medias res") will be elicited from a heterogeneous group of people while the constraints constantly change. Thus, a single elicitation method will not suffice – there is no panacea.

A clever combination of techniques is the road to success.

But before you begin eliciting requirements, you need to think about the appropriate technique. Our experience shows that the following factors have the greatest impact on the correct choice:

- The constraints of the project, especially the idiosyncrasies and capabilities of the stakeholders
- The distinction between conscious, unconscious and subconscious information
- The familiarity of the requirements engineer with a certain technique

Below, we will detail the different types of elicitation techniques and then (in section 5.4) rank them according to the above factors. When describing each technique, we'll delve into how apt it is to obtain conscious, unconscious and subconscious requirements.

5.2 The decisive factors

When eliciting requirements, it is of utmost importance to understand how requirements impact the satisfaction of your stakeholders. The Kano-Model, as originally presented by Dr. Noriako Kano in 1978, divides the features of a product into three categories [Sauerwein00]. These categories have differing effects on customer satisfaction with a product. Kano divides features into the following categories:

- *Basic features* are features which are taken for granted by the customer.
- *Performance features* are extras which have been asked for by the customer.
- *Exitement features* are features which the customer doesn't know or expect and will only discover about when using the product.

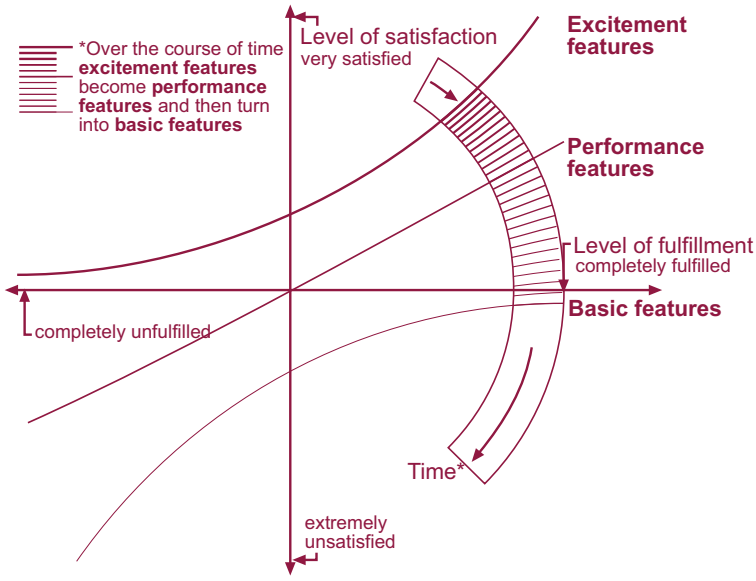


Figure 5.1: The Kano-Model shows what customers find satisfying

If the excitement features built into your product prevail on the market, other manufacturers will provide similar features. As a result, over time excitement features will turn into performance features and then into basic features. That means that in order to defend your position as the market leader, you have to stay creative.

Enthusiastic feature

In 1994, the capability of sending text messages was an exotic feature of certain cell phones, but soon the users discovered it to be a practical alternative to a regular phone call.

Performance feature

Since almost nobody positively did not want to send a text message, SMS became an often called for feature of mobile phones.

Basic feature

As users soon got used to SMS, nowadays it is implicitly expected that any cell phone bought will be able to send and receive SMS.

5.2.1 Unearthing basic features

= Subconcious knowledge

In figure skating, this would be the ability to skate at all..

The basic features are the features which the customer will expect your product to have, those your product or system must deliver. If these features are fully implemented, this does not entail customer satisfaction. Yet should one of these features be missing, massive dissatisfaction will be the result. Basic features of a mobile phone would be things as "ability to make a phone call", "reachable anywhere", "exchangeable battery" or "display and keypad on the same side". With our library example, basic features would be the ability to loan books, stop a loan procedure or the ability to log out anytime.

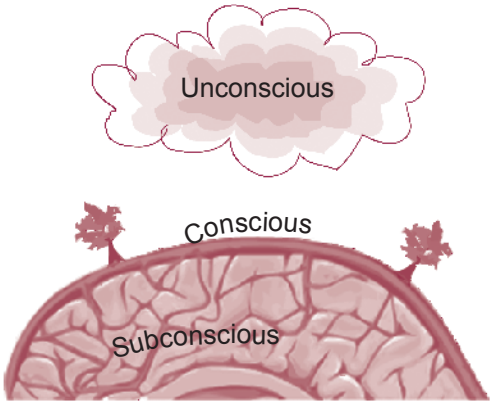
No matter whether you're developing a product for the open market or a custom-fit solution for a well-known client, you have to incorporate the basic features. If your halfway familiar with the domain at hand, you can probably lay them down on your own. But careful: even

the experienced requirements engineer is susceptible to assuming something is understood and failing to demand it explicitly.

If you're on unknown terrain, you'll have to rely on the knowledge carriers in the firm at hand or other sources, such as manuals to legacy systems. If both of these are no longer available, you'll have to analyze the code itself, mechanically test the system in order to gather requirements or use scenarios to discern what the system was really doing. When doing this, it is important to further scrutinize the information gained. Make sure not to adopt the information collected without further inspection. Demand to know whether each feature is still of use and what its economic benefits are.

System archeology

Oftentimes, unnecessary burdens of the past are draged along here = folklore



Another way to save time when determining the basic features is to delegate the work to someone else which you need not finance. More often than not, the basic features are just that, basic building blocks, which may be of use for other projects and purposes and thusly may be elicited by others. Who knows, maybe the requirements you're trying to elicit have already been gathered by some other project.

Figure 5.2: Different kinds of knowledge

= Conscious knowledge

Fullfilling wishes

5.2.2 Ticking off the performance features

Performance features are features which the stakeholder is aware of and which he is explicitly requesting. Providing these features creates customer satisfaction and is therefore desirable. If some of these features are missing, the customer will usually still accept the product, but his discontent will grow with every feature absent. In our library system, performance features would be details such as a display when books that have been loaned out will most probably be returned to the library, a notification when reserved books have become available for pickup or the possibility to reserve books from home.

The „program" in iceskating

You'll usually be able to gather the performance features first, since stakeholders usually express them explicitly. You'll thus be able to use questioning techniques to elicit them. In general, stakeholders tend to think performance features are all there is to a product. You might even take a look at the product the competitors are offering for some suggestions. But do make sure not to include too many performance features or utopist wishes by extremists, or else their implementation will cause problems. For those cases, double check the cost-benefit ratio. If you implement performance features in a system, the customer satisfaction rises. Somehow, customers like it when their wishes are fulfilled.

Basic features are usually forgotten and nobody knows any enthusiastic features.

= Unconscious knowledge

Identify yearnings

5.2.3 Compiling excitement features

The „freestyle" part in iceskating"

Excitement features are those attributes of a system whose value a customer will only understand when he has tried them out. For example, imagine being able to find your phone even when it's turned off. Perfect retention would be really handy when you're trying to remember that number or name. Or a bail out call, every time you're in a really boring meeting. In our library system, suggested reading subroutines would be really interesting – of course not in the typical manner (persons that read book x have also read book y) – but rather using criteria such as content, length, writing style, number of characters, location etc. Or a delivery service for books, preferably with the possibility to order a pizza from your favorite Italian at the same time, or you might even be able to add the Italian to the order, who would then read aloud from your book.

Whether a product becomes a bestseller and has a distinct advantage compared to the competitions products largely depends on excitement features. Timing is also very important. If your customer is in need of a highlight, it's a good time to mention such features. On the other hand, these features are especially hard to elicit, since they can't simply be obtained by questioning the customer.

In order to gather new and innovative ideas, it's best to let creativity run wild. Use such techniques as brainstorming or a change of perspectives during workshops with the stakeholders.

The ideas thusly collected must then be analyzed with regard to risks, feasibility, usefulness and their ability to delight. How much pontential an idea has can be determined using a "classic" elicitation technique, by simply asking the stakeholders how much (if at all) they like it. But it's not always the most technically challenging ideas that create customer satisfaction. The processes around a system, for example innovation and improvements in the areas of marketing, maintenance or learning systems, also offer room for excitement features.

Requirements are a socio-technical discipline

by Suzanne and James Robertson

We are often asked „What makes a good requirements analyst?" The short answer is willingness to listen, but it is worth looking a little further at the nature of the requirements activity to find a better answer to the question.

Requirements must be thought of as an activity that straddles the boundary between the sociological side of system development, and the technological side. On one hand we have people, with all their vagaries and fallibilities. On the other we have technology that demands a precise specification if the developers are to bring the best possible solution to the client.

There are several significant aspects to the sociological side of the activity. Firstly, the requirements analyst must identify and involve all the appropriate stakeholders to discover all requirements. Also consider that some stakeholders are too busy to pay proper attention, some don't know enough to supply the right requirements, and some think they know but don't.

What about the technological side of the fence? The skilled business analyst must know enough about the technology to know what is possible. People don't ask for things unless they know the things exist, or they have a good probability of being able to exist. So it falls to the business analyst to invent part of the system. If the analysts simply listened to their customers, then not only would each generation of system look pretty much like the previous ones, but few genuine advances would be made. Why is it important to see requirements analysis as a socio-technical discipline? Because software has become a commodity. There are too many people producing it, and too many people competing for your clients' software business. It is simply too risky to leave the requirements – the most important part of the development cycle – to chance.

James and Suzanne Robertson are the founders of the Volere requirements process, template and checklists. This acclaimed requirements technique is used by tens of thousands of organizations worldwide. Their careers have taken them to every continent and along the way they have collected an impressive portfolio of projects and industries. They can be reached through the Atlantic Systems Guild, a London, New York and Aachen consultancy and think tank. www.systemsguild.com.

Books:[DeMarco08], [Robertson06], [Robertson04], [Robertson98]

5.3 Elicitation techniques

Crystal balls and tarot cards

To educe knowledge, a multitude of techniques have been developed, all apt for different constraints. Our British colleagues, Suzanne and James Robertson [Robertson06] call their elicitation techniques "trawling techniques", a reference to fish trawlers casting out their nets to catch fish.

A good fisherman needs to know which net will catch which fish at what depth. Sometimes it's better to bring along a fishing rod, bait and patience. We've field-tested all the techniques on the following pages and can recommend each one.

Elicit unconscious knowledge

5.3.1 Creativity techniques

If you want to develop new, innovative ideas, you must let your creativity run free. Creativity techniques help think outside the box and give unusual ideas room to grow. When bringing them into play, make sure you establish an adequate environment for this creativity, else you

might just be creating creative chaos.

Creativity techniques are best suited for developing a first vision of a system, for gaining an overview and gathering innovative ideas. With their help, you may elicit unconscious excitement features.

But on a more abstract level - not suitable for tangible behavior or details..

Since you're applying your creativity, you also have a chance of discerning possible problems once the system is put to use or discovering innovative ways to better the system and thus gaining a gleam into future requirements.

Once good ideas have been engendered, you need to make sure they're also raised, are taught to walk and begin producing economic benefits. This can most readily be achieved by instigating an innovation process.

One such way to turn ideas into products has been delineated by Prof. Dr. Robert G. Cooper – the "Stage-Gate-Model" [Cooper02]. He likes to call it a "game manual" for new products. The Stage-Gate-Model® divides the innovation process into a set number of stages and gates. During the stages information is gathered, at the gates the results are checked and a stop-or-go decision is made. This way, you ensure that good ideas don't rot away in documents, or – the other extreme – that, although you have swarms of innovation projects, none make it to market.

Creativity techniques

- > Brainstorming
- > Brainstorming paradox
- > 6-3-5 method
- > Changing perspectives
- > Walt Disney-method
- > Bionik/Bisociation
- > Osborn's-checklist

Figure 5.3: The tools of the trade when working creatively

Brainstorming

Brainstorming is probably one of the best-known creativity techniques for groups. This technique was developed in the last few years of the thirties by Alex Osborn in order to increase the quantity and quality of promotional ideas.

During workshops mostly

Usually using mind maps

With a group of 5 to 10 people, ideas are gathered for a period of around 20 minutes and recorded by a host without further commenting – even if the ideas seem weird or altogether crazy. The participants use the ideas of other participants to engender new ideas of their own. Afterwards, the ideas are critically analyzed. A more detailed description of brainstorming and variants of this technique can be found in [Kellner02]. Through the use of hosting software and video conferences, it has become possible to conduct spatially distributed (electronic) brainstorming.

The host might have to do some quick writing, if the participants are extremely creative and produce tons of new ideas

Brainstorming works best if members of many different stakeholder-groups participate, the mood is good and participants encourage each other to produce new ideas. Then brainstorming can be a lot of fun. If the group dynamics aren't all that good, you need to make absolutely sure that even wild ideas are not criticized by other participants. If people fear their ideas will be the target of mockery and derision, most won't bother to participate. Thus good ideas might be lost. Be consequent and exclude troublemakers who won't quit their bashing – even after being admonished – from the brainstorming session.

Advantages of brainstorming

The advantages of brainstorming include that many ideas can be found in a relatively short period of time and that several people are refining each other's ideas. Because an uncensored collection of uninhibited ideas is made, new, never before thought of solutions may arise.

Disadvantages of brainstorming

If the group dynamics are jumbled or the level of dominance is very varied amongst participants, then brainstorming will not be effective, since the participants will interfere with each other. If the stakeholders are spatially distributed, brainstorming will require considerable effort, since the stakeholders need to be brought together in one place or need to be virtually assembled.

Brainstorming paradox

Brainstorming paradox [Kellner02] is a variation of brainstorming, where ideas detrimental to the goal at hand are collected. Using these results, measures are taken to prevent the impediments found from blocking the project or product.

Sometimes it's easier to find paradoxes than the real deal.

Brainstorming paradox is a technique which can be very much fun. Especially if the stakeholders don't know each other very well, this technique is more than adequate for braking the ice for example at the beginning of a workshop. Moreover, using this technique, you can obtain astounding results if you don't just incorporate members of the project team, but also include people that aren't stakeholders and don't have anything to do with the product. Why not invite the gatekeeper or a trainee? These outsiders will oftentimes be much more candid and merciless in their judgments, since they have no stakes in the project.

Advantages of brainstorming paradox

A huge advantage of this technique is that participants look at the matter at hand from a completely antithetic point of view and consciously determine which approaches will result in a negative outcome. Time and time again, we have found that such approaches with a negative outcome are evidently being developed for the project or have already been instituted. Brainstorming paradox helps you effectively assess possible risks and just like with regular brainstorming, you will find lots of new ideas in a relatively short amount of time.

Disadvantages of brainstorming paradox

Brainstorming paradox has exactly the same disadvantages as regular brainstorming.

6–3–5 method

Thus also foster creativity.

How down-to-earth or detached is up to the participants

The 6-3-5 method [Backerra07] is a written form of brainstorming. Six members simultaneously develop three ideas on their own and jot these down on a piece of paper. After a set period of time (usually 3 to 5 minutes) the papers are passed around in a circle. The participants then read the ideas their neighbor has written down and let themselves be inspired by these and denote three new ideas on the paper they got. This is done until everyone has had every paper once (5 times). Then the ideas are collected and analyzed.

Of course, you may vary this method in any way you see fit, for example 7-3-6 or 6-4-5..

This technique is easier to implement for the host than brainstorming, since the ideas are denoted, collected and grouped by the participants themselves. Therefore, the 6-3-5 method is apt for the less experienced host. Also, difficult group dynamics are not that much a problem with this technique, since every participant is equally involved, no silences arise and dominant participants need not be pacified.

6-3-5 is at its most effective when you point out that the goal of this method is to expand on the ideas which arise during the very first round. Of course, you shouldn't forbid new ideas categorically, or else you might just miss that grand solution.

Advantages of the 6–3–5 method

You can put this technique to use when group dynamics are complex, since the written form prevents the conflicts that usually arise out of discussions. You may even use this method with geographically distributed stakeholders, since it can be exercised via email.

Disadvantages of the 6–3–5 method

Compared to regular brainstorming, this method will produce fewer ideas, since the participants can't work together as actively. The process may also hinder creativity, for example when you can't finish thinking your idea through because time is over.

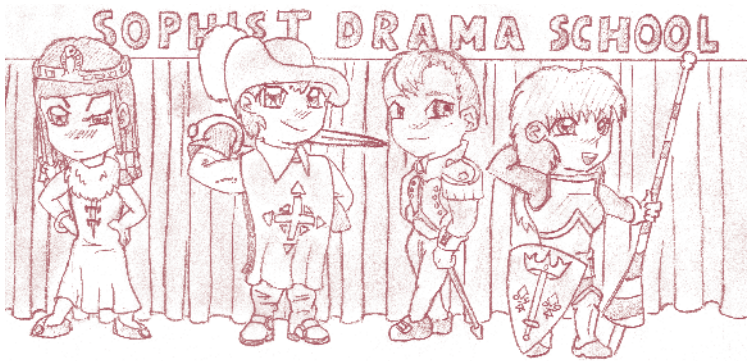
Changing perspectives (Six Thinking Hats)

There are a multitude of several-perspective-models designed to scrutinize a problem from different angles. The Six Thinking Hats technique by Edward de Bono is a very thorough variation with six perspectives, which may be exercised alone or in a group [DeBono06].

Try to force the points of view that none of the participants take to naturally

The participants are symbolically furnished with hats in different colors, which represent the perspectives from which the problem will be analyzed:

- Objectivity and neutrality (white): Facts and figures
- Personal sensations and subjective opinions (red): Feelings, fears, hopes
- Objective, negative arguments (black): Doubts, hesitations, risks
- Objective, positive traits (yellow): Chances, pros, goals
- New ideas (green): Random ideas, comparable to brainstorming
- Process control (blue): Hosting and guidelines



They need not be hats, colored cards will do the trick just as well. Using this technique, you'll get a very good overview of the entire project, especially if you use real roles from your project for the perspectives: you might give the white hat to a manager, the stakeholder with the red hat might utter the feelings of the librarian. It's important that you help the stakeholders slip into their perspective. Explain to them what embodying a certain perspective is all about, to get the most out of this exercise.

You might also take a closer look at something such as the development process of the system to be built if you give the hats to different roles that aren't normally filled at this point in time yet.

For example testers

In another interesting variation on this technique, you take a certain group of stakeholders and put the hats on different representatives of the same role. In our library example, you could for example take the role of "user" and look at it through different people's eyes (i.e. a child, a retiree etc.) in order to get a better understanding of the needs of the later users.

Advantages of changing perspectives

This method will allow even very obstinate and inflexible stakeholders to get a fresh perspective and think outside their normal frame of mind.

Disadvantages of changing perspectives

To many introverted or conservative stakeholders, changing perspectives seems like a very outlandish technique. It needs to be carefully introduced, not to scare the involved away. As a requirements engineer, you face the risk of being labeled as a bit of the esoteric/psychological type.

Walt Disney-method

Another several-perspective-model is named after Walt Disney, who is rumored to have had a different room for each perspective. This technique is particularly useful to develop very abstract and peculiar ideas. The Walt Disney-method [Dilts03] includes the following points of view, which are impersonated in spatially or temporally separated sessions:

- Dreamer and visionary: Fantasy, creativity, new ideas
- Realist: Feasibility and practicability
- Critic: Meaningfulness of an idea, weak points, negative aspects

The Walt Disney-method is based on the theory that in many people these three aspects hinder each other, or rather that one tends to dominate the other two. By explicitly separating these aspects and then concentrating on merely one, it becomes possible to really give room to all the facets of an idea and truly work creatively with it.

People with a good imagination are usually happy enough to just swap a seat, rather then change rooms.

The Walt Disney-method should be used by individuals and nicely combines a search for new ideas with an analysis from different viewpoints as seen in de Bonos Six-Thinking-Hats. It's important that you choose the rooms to be used with care and help the participant understand what each perspective is all about. The dreamers room should be something such as a meeting room without any technical machinery as for instance computers, but equipped with large board and colored markers. The daily workplace is usually ideal for the role of realist. But, as mentioned above, it's also possible to divide the standpoints temporally. The role of the critic could then be performed during a review meeting or a controller conference.

Advantages of the Walt Disney–method

You can use this technique to develop new ideas, assess them critically and test them for feasibility.

Disadvantages of the Walt Disney–method

Akin to the Six–Thinking–Hats technique, you need to make sure your stakeholders are ready to embrace the method.

Analogy–techniques (Bionics/Bisociation)

To develop solutions for a problem using Bionics, analogies from nature are used as a framework. The solutions provided by nature can then be applied to the original problem.

Think for example of the merger of two firms and compare it to the merging of two herds of animals. How long will it take, before the animals of both herds (the employees) are completely mixed? The leaders of the pack will start competing against each other and fight to create a new hierarchy. In a dangerous situation, for example if a predator attacks the herd, the animals of both flocks will act as one to increase their chances of survival. This pattern of behavior can be expected from the employees of both firms as well.

When using Bisociation, the analogies are not limited to examples from nature, therefore it is often easier to find a suitable analogy.

For both techniques it is important that all participants have at least a basic grasp and interest for the domain the analogy is from. The host thinks up the analogy, presents it as an exercise, but doesn't reveal which problem he is trying to tackle. That makes it easier to discuss the analogy, without the participants constantly thinking about how what they are currently saying will mean when transformed back to the business frame. Interestingly, most of the time, it is of no consequence whether the participants are stakeholders or outsiders. You can find more information on Bionics and Bisociation in [Kellner02].

Alternatively, you can have experts from the domain of the analogon illustrate the same and have the participants draw their conclusions from that expose.

Using this technique when everybody knows what the real problem is has no point, except if it's a triffle matter or an extremely good analogy.

Advantages of analogy–techniques

Complex problems or difficult–to–picture relationships become manageable through analogies. Changing the context helps tear down inhibitions. Experiences and solutions from other contexts can be brought to bear. We've used this technique very successfully to find unsuspected, creative solutions and oftentimes also discovered problems which we weren't aware of before.

Disadvantages of analogy–techniques

In order to be able to employ Bionics or Bisociation, you need a lot of time, since you need to build analogies first and then transform results back to your original problem–space. Faulty transformations of the results produced can lead to unsuitable solutions.

Osborn's-checklist

Test the user response to the product and derive suggestions for improvement.

May also be used on new developments, if a very concrete idea of the product exists..

Osborn's-checklist [Osborn79] is a particular type of questionnaire, which is given to representatives of the target audience, after a product, for example the legacy system, has been tested. This technique works best on tangible problems and physical objects.

Osborn's-checklist contains the following questions:

- *Put to other uses:* Could the product be used for a different purpose?
- *Adapt:* Is there anything else like this? Can we copy something off that?
- *Modify:* What can be changed? Can other functionality be implemented?
- *Magnify:* Can anything be added, improved, made more expensive?
- *Minify:* Can anything be taken away, made smaller or simplified?
- *Substitute:* Can the product or parts thereof be replaced?
- *Rearrange:* Is it possible to swap components, alter the pattern or sequence?
- *Reverse:* Can it be transposed? Can it be used to contrary ends?
- *Combine:* Can units be combined, blended, alloyed? Can it be used as a component of something else?
- *Transform:* Can it be compressed, liquefied? Can holes be made through it?

We make use of this technique quite often, because particularly the rather obscure questions such as to the aspect of transformation oftentimes lead to very creative suggestions. It's important that you, as the host, don't put too much importance on categorizing ideas or on making sure all the questions have been answered, because pressure and a rigid scheme will lead to frustrated stakeholders. If group dynamics are good, you can perfectly well use Osborn's-checklist for brainstorming sessions: simply ask one of the questions and have participants answer it orally or in written form, then expand on those ideas. If the group dynamics aren't all that well, you'll have to fend with the problems typical for brainstorms, so it's best if the checklist is used as a questionnaire.

Advantages of Osborn's-checklist

Osborn's-checklist is best suited when an existing product needs to be improved or augmented. Pre-specified, apparently obscure questions lead to unusual ideas.

Disadvantages of Osborn's-checklist

If the product is complex and multi-featured, it quickly becomes laborious to go through the checklist for each single feature. You should thus use this tool for the functionality of the entire system or for a few selected features which you find particularly interesting.

5.3.2 Surveillance techniques

Especially to obtain very detailed requirements

Not every stakeholder possessing important know-how can vocalize it. And usually, the most important stakeholders don't have the time to participate in the elicitation procedures (see [Beyer97]). In these cases, surveillance techniques are the tool of choice.

The requirements engineer observes the selected stakeholders – most of the time users of the system – while they work. He documents their workflow and uses it to delineate processes which the system will have to support. The stakeholders are either completely passive (with respect to the elicitation) – if they're only being watched – or they actively transfer their knowledge by showing the requirements engineer how it's done.

There's the danger that requiremetns will be caried from the legacy system to the system under development without further scrutiny.

When using surveillance techniques, risks are that the requirements engineer will be documenting outdated technological decisions and processes in dire need of improvement, since he is analyzing the current state of affairs. Here it's best to detail the essential requirements, to help abstract from technological decisions made in the past. The requirements engineer, as an outside observer, has a good chance of detecting inefficient processes and submitting suggestions for improvement. He has the necessary detachment, while the stakeholders usually repeat tasks that have become ingrained by years of habit, without paying too much thought. Do remember, though, that your presence changes the system. Only inanimate objects are unaffected by observation.

Even the statement that inate object are unchanged by observation is only partially true. A chair that you lift up for a closer inspection is subject to changes in temperature and potential energy.

Surveillance techniques are well suited to determine the basic features. An external requirements engineer will discern those basic features that most stakeholders take for granted or only know off on a subconscious level. Performance features might be observed, if they're part of the process or legacy system.

Figure 5.4: Looking rather than asking - for requirements engineers with a clear perspective

Field study

Including unconscious activities which are parts of complex processes

The requirements engineer denotes the activities of the stakeholder, their temporal relationships and the workflow. He may ask questions and demand to have unclear tasks clarified. Except when using creativity techniques, it's usually a must to have a look at the working environment of your stakeholders before beginning with the elicitation. It helps get a clearer picture of what it is your stakeholders are actually doing.

When doing field studies, you need to be extremely cautious: beware giving the impression that you are something of an overseer or inspector. Even more so when using video-documentation to support your field studies. In such cases, you'll absolutely have to talk to the stakeholder in question, and work out an agreement on the further use you'll be making of the footage, on how long you'll be saving it and if it's permitted to film at all.

Advantages of field studies

Conducting field studies is an excellent tool of choice if your stakeholders complete their tasks almost automatically (unconsciously), have a hard time vocalizing the same or aren't available for any extended stretch of time. The usage of this technique is also very effective when trying to determine deviations from the processes, since the requirement engineer can monitor many people and their activities.

Disadvantages of field studies

Activities that can only be observed under difficulty or not at all – such as an engine control – or activities that only happen on rare occasions, are not suitable for a field study. Also, stakeholders might feel uncomfortable because of the presence of a requirements engineer, thus corrupting results.

Apprenticing

RE-rule of thumb: If the requirements engineer sits in ambush, the stakeholder will be all hush and shush..

When employing this technique, the requirements engineer will be an apprentice to the stakeholders, who will coach him so as to understand the activities they carry out. This helps the requirements engineer get a very good impression of the instigated processes.

Most stakeholders will be quite happy to familiarize someone else with their line of work and will have no qualms taking the requirements engineer on as an apprentice. Using what he has learned, he can then go on to deduce detailed requirements for a supportive system. Moreover, the requirements engineer will inevitably experience scenarios which can later be used to construe test cases. Even potential error cases can be elicited at this point in time, since the requirement engineer – untrained – will commit mistakes which the veteran stakeholder doesn't make any more. You should be constantly aware, though, that you're learning about the current system and not detailing requirements for the system to be.

Particularly apt for work routines which are hard to survey

Advantages of apprenticing

Apprenticing is best suited if the stakeholders cannot vocalize their knowledge. Moreover, the stakeholder does not feel as if he were being inspected, rather he usually feels good since he is the "master" training the uncoordinated engineer. Particularly if the group dynamics are difficult, this technique offers psychological advantages, since the requirements engineer will obviously be having to admit weaknesses whilst learning and thus has the opportunity to exemplify how to admit not knowing something without a nimbus of fear.

Disadvantages of apprenticing

In critical environments, such as air traffic control for example, where mistakes can lead to serious hazards, apprenticing is inappropriate. When developing a product with an unclearly defined number of stakeholders, apprenticing is not the technique of choice, because of the time involved. Apprenticing is, for both, the requirements engineer more so and the "master" on a lesser scale, extremely time consuming and thus expensive.

To elicit conscious knowledge

5.3.3 Questioning techniques

Questioning techniques are the classics amongst the elicitation techniques and are based on asking the stakeholder about his desires and needs – which he, hopefully, will then express.

Sometimes, we're under the impression that the industry doesn't use anything but interviews.

Questioning techniques are suitable for requirements of any level of granularity, given that the stakeholder realizes the difference between these and is able to propound the same. Precise, non-functional requirements are usually hard to obtain through questioning techniques, since these requirements are usually hard to visualize for stakeholders.

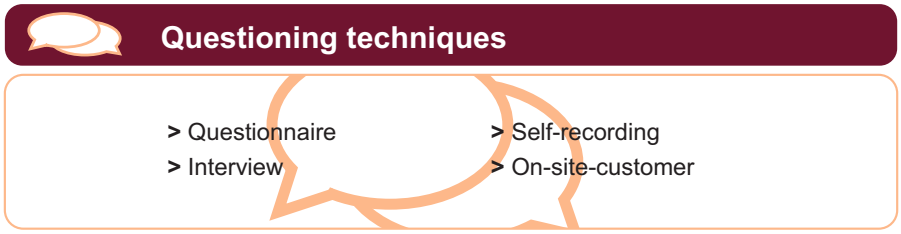


Figure 5.5: Elicitation of knowledge in the course of a dialogue

Questionnaire

Their evaluation can be automated

Most of the questions come from previously held interviews

A questionnaire is used to pose a number of open and closed questions, through which you attempt to gain the stakeholders knowledge. The questionnaire may be electronic or on paper and, for large audiences, might even be put online.

Couple the participation with incentives

Questionnaires are extremely well suited, when developing a product, to get a large number of people to evaluate planned functionalities and to get suggestions for improvement using the open questions. Make sure your interviewees are motivated enough to return a filled-out questionnaire. Also take care to put the most important questions first, so that only the rather irrelevant points are left unanswered if someone doesn't finish filling out the questionnaire.

Advantages of questionnaires

Using questionnaires, you can incorporate a large number of stakeholders into the analysis expending relatively little cost and effort, since the questionnaires can be distributed electronically and evaluated using tools.

Disadvantages of questionnaires

Questionnaires are unsuited to elicit implicit knowledge. Some types of requirements (such as non-functional requirements) can only be extracted in a limited fashion, since they're not quantifiable. Given that all questions are noted down, it's difficult to make further inquiries. Moreover, the way the questions are phrased may exert an influence on the answers given.

Interview

During an interview, the requirements engineer will ask one or more stakeholders predetermined questions and record their answers. Further questions which arise during the interview can be clarified on the spot (see [Leffingwell99]. The requirement engineer thus has the opportunity to elicit new requirements or uncover implicit requirements.

The questions must be formulated in a neutral tone and must not suggest certain answers. You may use a form to structure the interview. Before you begin, inform the stakeholder about which domain you will be asking questions and give him an estimate of the time it's going to take. Also, at the latest before beginning the interview, you need to clarify what use will be made of the answers and whether they will be kept confidential. Record the answers given and send the interviewee a protocol no later than within 48 hours of the interview. When the stakeholder now checks and approves the interview, you won't only be sure everything was understood correctly, but you're also signaling the stakeholder that his cooperation is of importance and you're taking him seriously. This usually motivates for further cooperation.

When beginning with the elicitation of requirements, personal interviews are well suited to delineate a first draft of requirements. To clarify details later on, the interview can also be conducted electronically. Of course, you probably won't be interviewing all stakeholders – thus pay attention to the correct choice of representatives.

Skype, ICQ, MS Messenger, ...

Advantages of interviews

The big advantage of an interview is that the requirements engineer can modify the course of the discussion individually as he best sees fit and can cater to the individual at hand. You get the chance to delve deeper if something is not completely answered or if more questions arise.

The physical presence of the requirements engineer vastly increases the chance that the questions will really be answered.

Disadvantages of interviews

Interviews with many stakeholders are time consuming. The correct choice of representatives is paramount.

The effectiveness of an interview largely depends on the experience of the requirements engineer. Supportive techniques and materials such as the SOPHIST-Set of REgulations or audio-tapes can considerably increase effectiveness.

Even more so than with a questionnaire, the phrasing of the questions will influence the answers, since now you've added facial expressions, gestures and tone of voice, all affecting the interviewee.

Listening, documenting and throwing out-target-oriented enquiries all at the same time, will be a bit much for even the most competent requiremetns engineer

Self-recording

An elicitation technique that quickly comes to mind is the authoring of activity reports by the person in possession of the required knowledge. The stakeholders document their requirements, change and optimization proposals on their own.

To increase the quality of results, you ought to introduce the participating stakeholders to some of the elicitation techniques. If you've got extremely motivated stakeholders on your

Der requirements engineer can use this as a basis and then go on to use other elicitation techniques.

hands, it sometimes pays off to familiarize them with some of the documentation techniques (see Chapter 8 "Documenting requirements"). Especially inexperienced stakeholders should be assisted, at least in the beginning, by an experienced colleague or a requirements engineer. Guidelines and templates (see Chapter 7 "Templates") help obtain standardized results of a higher quality.

Advantages of self-recording

The author is not influenced by the requirements engineer.

The stakeholder needs not explain his knowledge, but can formulate requirements right away.

Disadvantages of self-recording

The stakeholders usually only record conscious requirements.

The requirements delivered by stakeholders with lacking communicative skills need often be time-consumingly revised.

If the stakeholders are poorly motivated or have little time, the results will be lacking.

If many stakeholders need to be involved, evaluating the results will be very difficult, as variants will have to be consolidated and conflicts solved. Moreover, the specifications delivered are often of extremely dissimilar levels of granularity, style and wording, if you don't intervene in good time.

The idea comes from eXtreme programming.

On-Site-Customer

With this elicitation technique, there's always a representative of the stakeholders present as an on-site-customer of the developing team ([Beck99], [Beck00]). This permanent availability helps the involved clarify requirements and questions in short time. The stakeholder can, due to his presence, immediately test preliminary results or increments and detect and correct potential errors and misunderstandings. With the stakeholder continuously present, a very high level of granularity can be reached with the requirements.

Having an on-site-customer permits the unbureaucratic and efficient elicitation of requirements. If a critical system is being developed, the rapid feedback of the stakeholder present is of great advantage. He is an information source for general questions, for specific details he'll probably have to call on other stakeholders.

Advantages of an on-site-customer

Requirements are mainly elicited orally and therefore extremely quickly.

Disadvantages of an on-site-customer

To be granted a decisive, knowledgeable and communicative employee as a counterpart for the entire project runtime is usually difficult.

In addition, the on-site-customer must continually synchronize with those stakeholders not directly integrated into the project. If this doesn't occur, the disadvantages of this technique become quickly apparent: you may be getting fast answers - but these only represent the opinion of a single person. That in turn means that during the analysis phase no consolidation of stakeholder opinions is carried out, and once the system is rolled out, the risk is high that stakeholders that feel they were "left out" will try to sabotage the project.

Thus, an on-site-customer who is well integrated into the development team but has a bad standing with his colleagues can be very harmful..

To elicit unconscious knowledge

5.3.4 Artifact-based techniques

See [CPRE09] document-centered techniques

Today it is often the case that a legacy system is in use and all the people who knew how to cater to the processes "by hand" have long since left the company. The employees are not really knowledgeable about their specific domain, rather, they're simply users. They know which buttons to press and which steps to go through, but they can't tell you anything about the underlying logic of it all. This knowledge, this logic must thus be extracted from the system and its documentation.

Many businesses have lost the know-how embedded in large, complex legacy systems. If you've got the same problem: welcome to the club.

Paper and computer systems are oftentimes more patient than stakeholders -> well-suited for becoming acquainted with new topics

Artifact-based techniques reuse solutions and experiences embedded in successful systems. Also, only artifact-based techniques guarantee that the comprehensive functionality of the legacy system has been considered before the decision is made which parts of this functionality need to be included in the system under development. Using artifact-based techniques, you can figure out – to an arbitrary level of detail – how an existing system behaves. Thus you can determine all the basic features, which stakeholders might never mention, as well as implemented performance features. It makes sense to determine which parts of the legacy system will be reused and which parts need to be implemented anew.

Attention: unsuitable solutions might be carried from the legacy system to the new system

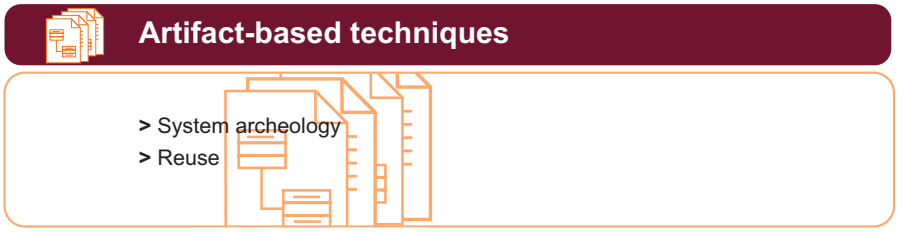
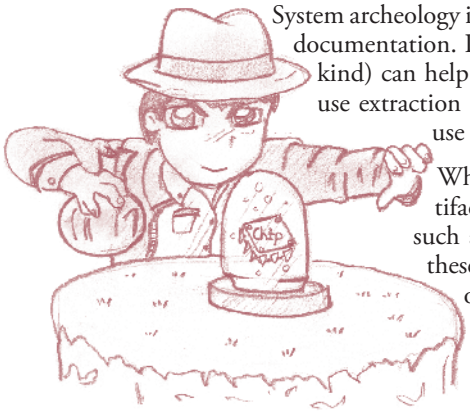


Figure 5.6: Stick to the facts, rather than doing some second-guessing

Artifact-based techniques ought to be combined with other elicitation techniques, in order to test the validity of the old requirements and to unearth new ones.

System archeology



System archeology is about uncovering requirements using an existing system and its documentation. Particularly a manual (or an online-tutorial or something of the kind) can help you quickly understand how the system behaves and you may use extraction techniques [John03] to reveal requirements. Other approaches use the manual to document requirements [Rupp04].

When employing system archeology, always begin with those artifacts that illuminate the functionality of the system most clearly; such as the user manual or old test cases. If, after having processed these documents, open questions remain, you'll have no other option than to descend into the depths of the legacy system and take a look at the code.

Advantages of system archeology

When analyzing an existing system or parts thereof, you can usually be sure that none of the already implemented functionality will be forgotten.

Disadvantages of system archeology

System archeology is a very complex and time-consuming technique, which can only deliver the scope of operation of the legacy system.

In a quickly developing market where changes are common, there is no point to conducting system archeology, since most of the functionality will have to be elicited anew.

Further problems may surface if the surviving documentation is of inferior quality or is out-of-date (the system was modified after the documentation was printed).

Reuse

Will reduce costs ... when employed correctly

Have you developed a similar system before? Then you may reuse requirements and other artifacts from the development of the past project. You'll need to examine all the artifacts you can still get a hold of. The specification document will be of particular interest. Look out for processes mentioned more than once, such as "searching" or "saving", in order to identify the most probable candidates for reuse.

In the best of worlds, you've already been using a knowledge database, where you've deposited requirements on a suitable level (for example the use case level) for easy retrieval (see Chapter 16 "Reuse"). It's easiest to reuse requirements if you've generalized them beforehand and now only need to adapt them to better suit the current project. In order to be able to reuse requirements and their derivatives, they need to be well documented, which means extra work in the first project. These costs can usually not be budgeted to the project costs and it's therefore difficult to advocate such practices. What really helps in these cases is a corporate culture where preventive arrangements for quality increase and cost reduction are instituted. Not relying on knowledge gained in the past has been given a name: the Not-invented-here-syndrome refers to the common practice of failing to investigate whether the current set of problems hasn't been solved in the past. Rather, elicitation, invention or description is performed again. A reward program can help motivate employees document knowledge in a reusable form and check whether the knowledge currently needed hasn't been created before.

A mandatory, realistic reuse-quota will force people to inspect existing material

Advantages of reuse

You save on costs if you reuse requirements, since these have already been elicited and quality tested. The time and effort for reviewing and correcting these can thus be greatly reduced. Possibly, there's even more information to go with these requirements such as test cases or parts of a model.

Disadvantages of reuse

The biggest problem when practicing reuse is finding the right requirements. Oftentimes the quality of the old requirements isn't as high as it ought to be. A reuse without further quality control can lead to defects being carried across systems.

5.3.5 Auxiliary techniques

In order to increase the effectiveness of the elicitation techniques described above and to increase the quality of the elicited requirements, auxiliary techniques may be combined with elicitation techniques. Which combination is most suited to minimize the weaknesses of a certain elicitation technique has already been provided in the description of that technique. We're only listing the most common auxiliary techniques here, find more on www.SOPHIST.de.

Figure 5.7: Was sonst noch alles beim Erheben hilft

SOPHIST–Set of REgulations

Actually, the SOPHIST-Set of REgulations should be a parser running in the back of your mind..

Neuro-linguistic programming (NLP) offers a number of elements, models and techniques designed to foster target-oriented communication. In order to better understand an utterance, NLP makes use of a so-called "meta-model" of language (see [Dilts03], [Bandler75]). This model makes it possible to detect implicit requirements and allows for a more precise definition of ambiguous or plurivalent requirements. This technique could well be called a tool, since it may be used during workshops, reviews … whenever language plays a role. Chapter 6 "The SOPHIST–Set of Regulations" delineates how requirements are analyzed in such fashion.

Make use of the SOPHIST–Set of Regulations to increase the level of quality of existing requirements or to generate further questions when interviewing stakeholders.

Advantages of the SOPHIST–Set of REgulations

Through the use of the SOPHIST–Set of REgulations, requirements denoted or vocalized by the stakeholder can be analyzed to uncover gaps and implicit assumptions, thus increasing the level of quality of the requirements drastically.

Disadvantages of the SOPHIST–Set of REgulations

The correct and efficient use of the SOPHIST–Set of REgulations requires some experience and therefore must be taught and practiced. It takes time to get up to speed on the method and during the practice phase, going through all the rules for each requirement can be quite time–consuming.

Workshop

Complex processes with many stakeholders require a joint compilation and consolidation of results by the relevant stakeholders. In a workshop, stakeholders with the required domain knowledge and decision-making authority come together in order to produce concerted requirements. Already gathered requirements might be sorted according to relevance, bundled according to content and substantiated in such a workshop, for example. Or the workshop serves as a platform to discuss open questions. A workshop progresses according to a predetermined agenda and has set rules for whose abidance the host is responsible.

Who has been invested with adequate authority

There are hundreds of tips and tricks for conducting a successful workshop (mail around a structured agenda; define the target audience beforehand; document results; …). To name them all here would exceed the scope of this work. You may find further suggestions and information on the matter in [Robertson06], [Leffingwell99] and [Wiegers99].

Advantages of workshops

Direct communication promotes mutual understanding and the willingness to accept compromises. Furthermore, it allows teams to acquire concerted information.

Disadvantages of workshops

Negative group dynamics can lead to ineffectual workshops. What's more, workshops are not feasible given large numbers of stakeholders, spatially distributed stakeholders or poor availability of participants.

Mind mapping

Mind mapping was developed in the seventies by Tony Buzan [Buzan05]. The method is designed to systematically order and structure ideas and terms according to congruity. It's actually more of a documentation technique, but its use encourages creativity.

Beginning with a central theme, branches carrying information are drawn, which in turn lead to other branches with more detailed information. Every branch carries headwords. By using symbols for important elements or relationships and colors for different levels, an easy-to-understand structure is assembled.

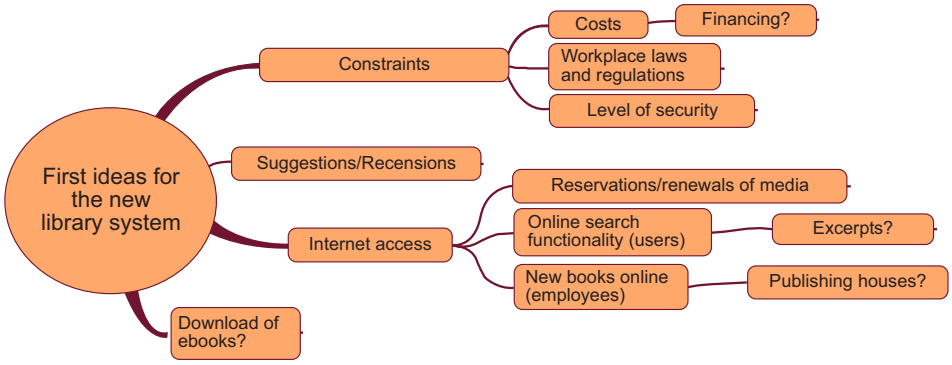


Figure 5.8: A mind map as a knowledge trove

There are excellent tools for quickly creating mind maps..

Mind maps serve to structure ideas and visualize the relationships between them. They may also be used to document conversations or thoughts. It's best to give every participant a marker and to build the mind map as the discussion progresses, for example using a whiteboard or flipchart. Since the ideas become immediately visible for everyone and there is no need for formalism, everybody has the opportunity to develop further associations and add them – at best with a short oral explication.

Advantages of mind maps

A mind map is well suited to visualize thoughts, structure these and document them. It encourages creativity.

Disadvantages of mind maps

A correct interpretation of the resulting diagram can usually only be made by the author and other participants. Thus, mind maps are not apt for storing information for third parties over extended periods of time.

Audio recordings

= questioning techniques

The data that can be gained from oral elicitation techniques usually cannot be documented comprehensively – due to the speed with which we speak. A subsequent documentation from memory is usually also fragmentary. To make sure no information is lost, an audio recording can be made. The recording device will record all statements made. Using this recording, the statements can then be put in writing.

It is crucial you seek the permission of the stakeholder to record the interview beforehand. While at it, delineate how confidential the recording is, how and what for it will be used and how long it will be stored before being destroyed.

Advantages of audio recordings

Audio recordings vastly increase the speed with which interviews can be carried out, since the statements need not be written down. At the same time, no information is lost and even casual remarks can be taken into account.

Disadvantages of audio recordings

Insecure people might feel threatened by the recording, since it may later be used to reveal mistakes or lack of knowledge.

Processing audio recordings takes a lot of time.

Video recordings

Video recordings can assist you when using questioning techniques, for example to record not only what has been said during a workshop, but to also catch the non-verbal reactions of the participants.

Video recordings are also used to document activities which are complex or are carried out at high speed. If the requirements are not documented in written, the video can serve as documentation.

Another application for video recordings is the assessment of simulation models. Stakeholders are confronted with the simulation and filmed while using the system. By carefully analyzing the reactions of these users when interacting with the system, you can evaluate the usability of the system. Hence usability requirements can be made mensurable.

The permission to film is a prerequisite here too. Clarify beforehand how sensible the recording is, the uses to be made of it and the time it will be retained.

Advantages of video recordings

Surveillance techniques become more efficient if the workflow is recorded.

The ability to record non-verbal information during questioning or creativity techniques is another positive aspect of this auxiliary technique.

Disadvantages of video recordings

Stakeholders might feel monitored and thus might oppose recordings.

The knowledge that one is being watched can lead to slightly abnormal behavior.

Processing video recordings is time consuming.

Video-based requirements engineering

by Oliver Creighton

Oftentimes even experts quickly reach the limits of their capability to express thoughts when trying to specify requirements. Especially the precise and fast communication of complex processes and interactions between humans and machines is problem-laden, because it entails visualizing an innovative system in a factual context before this system ever comes to be.

Ostensive media are therefore essential to facilitate sound decisions on the scope of a system, as well as to transport customer demands through the different stations of the development process without distortions.

In the pertinent literature, we find suggestions on how to handle such problems when involved with scenario based methods, use cases, user-centric or participatory design. Videos are well suited to be used to this end. The often claimed downside that tape-based video technology is too unwieldy to be of use, can no longer be upheld in these times of YouTube and digital cameras.

In order to facilitate video-based requirements engineering, we've defined a process model which is augmented by a partially self-developed toolbox. This model dictates that demonstrations or improvised acting using requisites are filmed in their actual contexts, rather than going into the usual interview game. The resulting film can then be transformed into diagrams using a film-language with computer assistance. The product is a formal model of the problem space. The feasibility of this approach, in particular concerning the required time and effort, has been well proven in workshops and research projects.

Multimedia is entering the international requirements engineering scene, as periodic workshops on the subject are being held (www.mere07.de).

Dr. Oliver Creighton (oliver@creighton.de) is the requirements engineering program manager with Corporate Technology of Siemens AG. He is responsible for coaching and vocational training and the development of innovative requirements engineering technologies. He co-authored the book "Open-Source-Software" (2004, Springer). Furthermore, he does research on knowledge models and communication through films.

Use case modeling

Use cases [Cockburn00] make it possible to document an abstract outlook of the system, while avoiding getting lost in the complexity of process details. Used in workshops, they're an excellent way to elicit knowledge and document it at the same time.

This kind of analysis focuses on the business events which are part of the system and resulting actions. The use case represents a service the system provides, with an outcome which is of value to the user (see Chapter 8 "Documenting Requirements").

Employ use cases during workshops. Model the use cases found on a pin board during the discussion with the team. Document the most important aspects of each use case on a flip chart. By doing this, the team and you can jointly come to an agreement concerning the most important functionality of a system.

Advantages of use case modeling

Use cases impose a functional view and help logically dissect the range of functionality a system offers. If the system is large and complex, it helps structure the analysis.

Disadvantages of use case modeling

There are no clear cut rules for describing use cases in prose. If several teams work in parallel, you must absolutely agree on a few rules to make results compatible.

Reduction to the essence

A prerequisite for reuse

When eliciting requirements, there's always the danger that stakeholders will describe work routines including the currently implemented technological solutions and that the resulting requirements will thus be technologically biased. This makes an improvement of the system more difficult, because old decisions will be embedded in the requirements. Furthermore, stakeholders frequently devise solutions that lead to an unnecessary complexity of the system. Before you begin with the implementation, you ought to reduce the processes to their essentials, in order to purge outdated solution-approaches. Beware of readily using the suggestions the stakeholders make without further appraisal, because if you do, you'll be building an expensive and complex clone of the legacy system.

We call this problem „folklore".

In order to find the essence, you must begin by defining concrete sequences of activities and dependencies, and then generalize these. The outcome will be generalized, essential requirements. An example: the librarian describes which fields he must fill out on a certain form when cataloguing a new book in minute detail. If you use these statements without further scrutiny, the new system might require exactly the same steps. New technologies, such as Barcodes or RFID-chips which are machine-readable, might never become part of the system.

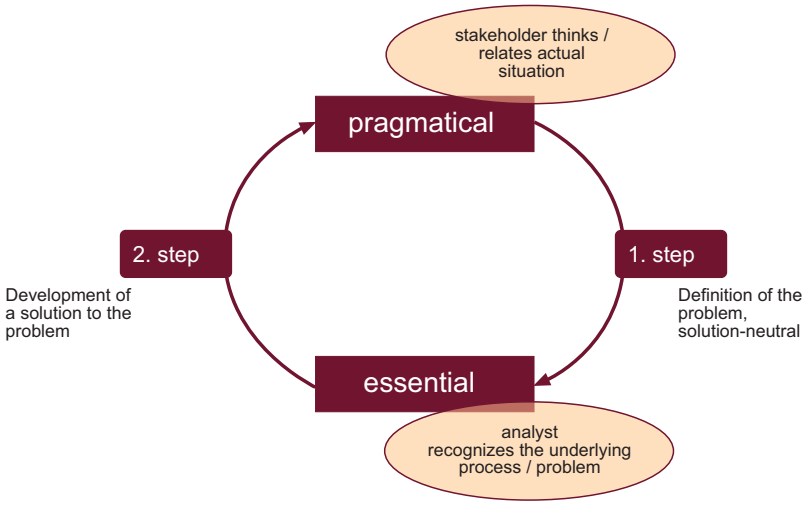


Figure 5.9: From pragmatic to essential and back

Aside from the fact that you're gaining the freedom to find the very best possible solution for a certain implementation, reducing to the essence gives you the opportunity to recognize universally valid rules which you might reuse in other projects.

Advantages of reducing to the essence

By reducing to the essence, the overall complexity of the system specification is lessened. Discussions will not drift off into the realm of non-issues and the underlying problem stays in focus.

Disadvantages of reducing to the essence

Sticking to the essential ifs oftentimes very difficult for stakeholders, because they are deeply involved with the current way to do things. Most of the time, it is necessary to bring in an external requirements engineer with a good aptitude for abstraction.

When reducing requirements to their essentials, part of the information is lost (the non-essential part). Contemplate whether you might want to document this part too.

Divining requirements

The requirements engineer as a miracle worker

If the requirements engineer has enough experience and is quite knowledgeable in the domain in question, he may assemble elemental requirements without involving the stakeholders. Using the information about the system he has at hand, he creates requirements based on assumptions. Later, these requirements can be validated by the stakeholders during, for example, a review session.

Used frequently;
find the antipole to what's already there:
loans - returns
dispose off - open file

This is a common technique in today's system development. In conjunction with prompt reviews, this technique can increase the effectiveness with which requirements are gathered. Sadly, the review is frequently delayed until the product is implemented. This approach is obviously less-than-ideal, since requirements engineering is about the desires of the customer, not the fantasies of some engineer or developer.

Advantages of divining requirements

Since the stakeholders are only involved in the review of the requirements, this technique can be very efficient and help gather plenty of detailed requirements.

Disadvantages of divining requirements

Since the stakeholders aren't involved from the start, there's the risk that the requirements will not concur with their needs.

The review is usually a lengthy affair, since every single requirement has to be scrutinized.

5.4 Putting it to practice

Stakeholders cannot formulate essential, uncluttered requirements.

Figure 5.10 shows which elicitation technique is best suited under which constraints. The assessment that this table is based on is the result of our years of experience and feedback from colleagues – but of course it's not supposed to be a dogma. If you've made different experiences on your own, please feel free to adapt the table as needed. You'll find an editable copy on www.SOPHIST.de.

Legende:

–	not recommended
0	no influence => may be used
+	recommended
++	highly recommendable

	Brainstorming	Brainstorming paradox	6-3-5 method	Changing perspectives	Walt Disney-method	Analogy-techniques	Osborn's-checklist	Field study	Apprenticing	Questionnaire	Interview	Self-recording	On-site-customer	System archeology	Reuse
Human constraints															
Stakeholders lack motivation (to participate actively)	–	–	–	–	–	–	–	+	–	0	+	–	–	++	++
Communication skills lacking	–	–	–	–	–	–	–	++	++	–	+	–	–	++	++
Ability to abstract defficient	–	–	–	–	–	0	–	++	++	0	+	–	–	++	++
Many different opinions	+	+	++	+	0	+	+	++	++	+	0	0	–	0	0
Imbalance in power between parties involved	–	–	+	–	0	0	0	0	0	0	0	0	0	0	0
Group dynamics askew	–	–	+	+	+	0	0	0	0	0	0	0	0	0	0
Organizational constraints															
Development for a complex market	++	+	+	+	++	+	+	–	–	++	0	0	0	+	0
Fixed, tight budget	++	++	++	+	+	–	+	+	–	–	+	+	+	–	++
Stakeholders physically far apart from each other	–	–	0	–	0	–	0	0	0	++	0	+	–	0	0
Stakeholder availability poor	+	+	+	–	–	–	+	+	–	+	++	–	–	++	++
High number of stakeholders	+	+	–	+	–	0	0	0	–	++	0	–	–	0	0
Technical constraints															
Highly critical system	0	0	0	+	0	0	0	++	–	+	+	+	++	++	+
System has a large scope	0	0	0	0	0	–	–	+	–	–	+	–	+	+	+
No previous experience in the domain	0	0	0	0	0	0	0	–	+	–	–	0	+	+	+
Trying to find rough requirements	++	++	++	+	+	+	++	+	0	+	++	+	+	–	0
Trying to find detailed requirements	+	+	+	+	–	0	0	+	++	–	+	+	+	++	+
Non functional requirements	0	0	0	0	0	+	+	0	+	–	+	0	++	+	+
Very complex system	0	0	0	0	0	+	–	–	–	–	+	–	+	+	+

The requirements engineer is no domain expert.

Stakeholders are unknown or unavailable

Figure 5.10: The relationship between elicitation techniques and project reality

If a technique is listed as "not recommended" (-) under certain constraints, you shouldn't use it. Entries marked "highly recommendable" (++) denote techniques that will work very well under the given circumstances. "Recommended" (+) may well be used. If a technique is marked with a "no influence" (0), the corresponding constraint doesn't influence the technique, but there's usually a better choice.

To determine a suitable technique, follow these four steps:

1 You'll have to resolve which kind of knowledge you're looking for, before choosing a technique. This is where the Kano-Model from section 5.2 will help you. Based on it, the following distinctions can be made:

- Basic features can best be elicited using the surveillance and artifact-based techniques described in sections 5.3.2 and 5.3.4.
- You can simply inquire about performance features (use the questioning techniques from section 5.3.3).
- Excitement features must be compiled using the creativity techniques from section 5.3.1.

Subconcious knowledge

Conscious knowledge

Unconscious knowledge

2 The next step is to analyze the influencing factors. Go to the table and mark 3 or 4 of the constraints that you deem most prominent in the project under question.

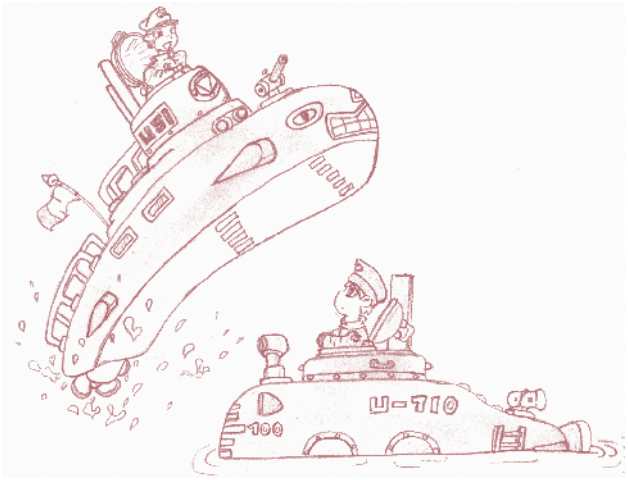
3 Now look through the table for the techniques you chose in step 1 and find those that score highest under the constraints you chose in step 2.

Since you'll usually have to elicit basic, performance and excitement features for a system, you'll end up with a mix of elicitation techniques, which will help you minimize risks.

Since no elicitation technique is without blemish, it's typically a good idea to shop around amongst the auxiliary techniques. You'll generally find one or two methods that will help you make the most of your chosen elicitation techniques.

4 If the elicitation process has somewhat ground to a halt, or you need something to ease up a deadlocked workshop, use the rather more outlandish creativity techniques. With an extravagant creativity technique, you may just be able to break open entrenched structures by arousing some attention and advertising for requirements elicitation.

Last resort



But never forget one thing: creativity can't be forced. Sometimes a little break or a change of location will work much better than the most bizarre elicitation technique.

5.5 The accomplished clairvoyant

Any similarities to events, persons living or dead, are purely ....

In this section, we'll take our library example and show which elicitation techniques we would use. We'll analyze under which constraints and why these techniques make sense.

There's this idea buzzing around, that the system should provide reading suggestions based on books loaned in the past. The proposals range from simply listing other books by the same author, to an analysis of what comparable readers read and deducing suggestions from that (everyone probably is familiar with that one from a well-known online bookstore). But we're looking to gather other innovative ideas. The following influencing factors were identified as most relevant for this project:

- Problematic group dynamics
- Many contrasting opinions
- The availability of the stakeholders is poor
- The requirements engineer (us) doesn't know the domain very well

Ok, with that being said, let's simply go through the four steps described above:

1 Since we're looking for new ideas, creativity techniques are the method of choice.

2 Mark the relevant constraints (see figure 5.11).

3 Choose the elicitation techniques based on steps 1 and 2. In this case, that would be the creativity technique which scores highest on the condensed table (figure 5.11). We'd recommend using the 6-3-5 method, since – as a written variant of brainstorming – it's best suited to handle challenging groups, and due to its structured approach, we're hoping to capture many different thoughts and opinions.

4 Use an auxiliary technique to keep the weaknesses of the elicitation technique in check. We'll use the SOPHIST Set of Regulations to cleanse the results of defects and identify any gaps or ideas that weren't thought through. Then we'll fill these gaps jointly with the author responsible.

Legend:

–	not recommended
0	no influence => may be used
+	recommended
++	highly recommendable

	Brainstorming	Brainstorming paradox	6-3-5 method	Changing perspectives	Walt Disney-method	Analogy-techniques	Osborn's-checklist	Field study	Apprenticing	Questionnaire	Interview	Self-recording	On-site-customer	System archeology	Reuse
Human constraints															
Many different opinions	+	+	++	+	0	+	+	++	++	+	0	0	–	0	0
Group dynamics askew	–	–	+	+	+	0	0	0	0	0	0	0	0	0	0
Organizational constraints															
Stakeholder availability poor	+	+	+	–	–	–	+	+	–	+	++	–	–	++	++
Technical constraints															
No previous experience in the domain	0	0	0	0	0	0	0	–	+	–	–	0	+	+	+
Trying to find detailed requirements	++	++	++	+	+	+	++	+	0	+	++	+	+	–	0

Figure 5.11: Condensed table for our example

5.6 Clairvoyant's how-to

Elicitation techniques checklist

1. Based on the of kind knowledge you're trying to elicit, choose the appropriate type of elicitation technique
2. Distinguish 3 or 4 of the most prominent constraints
3. Choose the highest scoring elicitation technique based on steps 1 and 2.
4. Compensate for weaknesses in the elicitation technique by using a suitable auxiliary technique