

Requirements Elicitation

Notes by mainly Jo Anne Atlee,
with modifications by Daniel Berry
dberry@uwaterloo.ca

Fall 2004

Outline

- Most Important Aspect of RE
- Who are the stakeholders?
- Main Task — Examine Project Viability

- Job of Requirements Analyst
 - Understand problem from each stakeholder's point of view.
 - * Review documentation
 - * Observe current system
 - * Questionnaires and Interviews
 - * Apprenticeship
 - Extract the essence of the stakeholders' requirements
 - * Interpreting stakeholders' descriptions of requirements
 - * Building models
 - Invent better ways to do the user's work
 - * Ask why documented requirements are desired
 - * Consider giving the user more creative control over his or her
 - * Brainstorm to invent undreamed of requirements
 - Negotiate a consistent set of requirements
 - * Key Aspects of Requirements Negotiation
 - Record results in an SRS

- What Can Go Wrong in Elicitation and the SRS?
- Other Techniques
 - PIECES
 - Social and Organizational Factors
 - Ethnographic Analysis
 - Joint Application Design
 - Names and Norms
 - Gause & Weinberg Ideas

Most Important Aspect of RE

What is the most important aspect of the requirements process?

The two leading candidates:

- the Software Requirements Specification (SRS) document
- the process of negotiating requirements that are agreed to by all stakeholders.

Elicitation and Brainstorming

Daniel M. Berry

Definition

“to elicit”

means

“to bring out, to evoke, to call forth”

In this case, information pertaining to requirements

Purpose of Elicitation -1

The purposes of elicitation is to get information about:

- **the domain model from which the requirements are written**
- **the requirements from which system is developed**

Purpose of Elicitation -2

You must get information out of clients' minds without damaging the clients or their minds!

Many times this information does not come out easily.

The clients do not know it themselves.

The clients do not want to let it out (subconsciously).

Purpose of Elicitation -3

Elicitation is a human activity involving interaction between human beings:

- **clients**
- **users**
- **systems analysts**
- **systems developers**

Purpose of Elicitation -4

If you cannot do the human interaction right, you ain't gonna be able to elicit, no matter what technology and methods you use.

Technology and methods might help, but they can also get in the way.

Skills -1

The skills needed for elicitation are:

identifying contexts

spotting ambiguities

interviewing

brainstorming

facilitating

getting people to open up

spotting equivocation

inculcating guilt

Only the first two are not human interaction!

Who are the stakeholders?

- Client — person paying for the software to be developed
- Customer — person who buys software after it is developed
- Users (of both the current and future systems)
- Domain Experts — experts who know the work
- Software Engineer — technology expert
- Inspectors — experts on government and safety regulations
- Market Reseachers
- Lawyers
- Experts on Adjacent Systems

Client — person paying for the software to be developed

This is the ultimate stakeholder. By paying for development, the client has the last say in what the product does, how it does it, and how elaborate or sparse it is. In some sense, by being willing to pay for the development, the client demonstrates just how interested he or she is in the product.

If you are developing in-house software, the client is probably the manager of the product's users — since his or her employees will be the primary beneficiaries, it is reasonable for him or her to pay for the project

If you are developing software for the mass market, then the client may be your marketing department.

Customer — person who buys software after it is developed

You have to understand the customer's needs well enough to build a product that he or she will find useful and buy. Sometimes the customer and the user are the same; othertimes, the customer is an office manager who buys software for his or her staff.

For what requirements will he or she pay? Which are trivial or are excessive?

The customer should always be represented by a stakeholder who is active on the project; if there are many customers, there needs to be a customers' representative.

Main Task — Examine Project Viability

One of the first tasks is to learn enough about the project to decide whether or not it makes good business sense to begin doing the project. For some reason, it is very difficult to cancel a project once it is underway. The more resources that a project has consumed, the harder it is for it to be cancelled. Most managers would rather stick with a dead-end project, than cancel it — even if it is more expensive in the long run to stick with it. To cancel a project is to admit error, which many managers are loath to do.

Ed Yourdon has written all about Death March projects.

Determining viability requires examining the product's:

- purpose,
- business advantage,
- costs vs. benefits,
- feasibility,
- scope,
- required resources,
- requirements constraints, and
- risks.

Job of Requirements Analyst

1. Understand the problem from each stakeholder's point of view.
2. Extract the essence of the stakeholders' requirements.
3. Invent better ways to do the user's work.
4. Negotiate a consistent set of requirements.
5. Record the results in an SRS.

First some detail and then some more detail...

Understand Problem

- Why Analyze Existing System?
- Steps in Analysis:
 - Review Documentation
 - Observe Current System
 - Questionnaires and Interviews
- Apprenticeship

Review documentation

Review all available documentation. If there exists an automated system, review its documented specifications and user manuals. If the existing system is a manual system, review any documented procedures that the workers must follow.

The goal is to gain knowledge of the system before imposing upon other people's time, before bothering the stakeholders.

Observe current system

Documentation rarely describes a system completely, and it often is not up to date. The current operation of the system may differ significantly from what is described.

Besides, no matter how bad a reputation the existing system has for doing the work, the system is not worthless. It contains a lot of useful functionality that should be included in any future system. The objectives of observing the current system is to identify what aspects to keep and to understand the system you are about to change.

Questionnaires and Interviews

Questionnaires are useful when information has to be gathered from a large number of people, particularly users.

Questionnaires are useful also when the answers to questions need to be compared or corroborated.

There are a couple of points about questionnaires and interviews I want to stress.

- interview all stakeholders

A common mistake is to interview only the client and the user and to neglect the other stakeholders, who may have definite views about what the system should do.

Common Interviewing Mistakes

As this is labour and time intensive (and therefore costly), you don't want to diddle about. These are the four most common mistakes:

1. Not interviewing all of the right people.

Different stakeholders have different points of view. Be careful to talk to everyone appropriate.

2. Asking direct questions too early.

e.g., *Designing a transportation system:*

→ *How many horsepower do you need? (direct)*

→ *How many people? How far? How fast? (indirect)*

e.g., *Camera design for novice photographer:*

→ *How important is control over shutter speed and aperture? (direct)*

→ *Will you be taking action shots, still shots, or both? (indirect)*

Apprenticeship

Apprenticing is a wonderful way to observe the real work.

Apprenticing is based on the idea of masters and apprentices. In this case, the RA is the apprentice and the user is the master craftsman. The apprentice sits with the master craftsman to learn the job by observation, asking questions, doing some of the job under the master's supervision.

Brainstorming -1

Brainstorming is already part of our culture, but beware of bad brainstorming.

A bad brainstorming session is a brainblizzard because it freezes your brain, leaves you under mounds of snow, and leaves you cold

We will give rules for brainstorming that help avoid the brainblizzard.

Brainstorming

- When you have no idea, or too many ideas, sit down and thrash it out ... but with some ground rules.
- Most useful early on, when terrain is uncertain, or when you have little experience, or when novelty is important.
- Who participates?
 - developers, domain experts, end-users, clients, ... just about any stakeholder can take part.
 - Often, software development companies will have special-purpose “ideas-guys”^a who lead or attend these meetings, but may not participate beyond this stage.

^aCould be female or male.

Brainstorming

- Want to hear ideas from everyone, especially unconventional ideas.
 - *keep the tone informal and non-judgemental*
 - *keep the number of participants “reasonable”, if too many, consider a “playoff”-type filtering. Invite back most creative to multiple sessions. or it’s too hard to be heard (only the loud will prevail).*
- Creativity to be encouraged, which means:
 - Choose good, provocative project name.
 - Choose good, provocative problem statement.
 - Get a room w/o distractions, but with good acoustics, whiteboards, coloured pens, provide coffee/donuts/pizza/beer
 - Provide appropriate props/mock-ups (*e.g.*, ComfyCrate)

Brainstorming

First, must designate two (different!) people for special roles:

1. Scribe — Role is to write down all ideas. May also contribute. May ask clarifying questions during first phase, but not critical questions.
2. Moderator/leader — Two schools of thought on this:
 - (a) Traffic cop — enforces “rules of order”, but doesn’t throw his/her weight around otherwise.
 - (b) Agent provocateur – Assumes more of a leadership role, comes prepared with wild ideas and throws them out as discussion wanes. May also explicitly look for variations and combinations of other suggestions. Not a “philosopher-king”. Also acts as traffic cop.

Brainstorming

Part I — The Storm

- Goal is to generate as many ideas as possible.
- Quantity, not quality, is goal at this stage.
- Look to combine or vary ideas already suggested.
- No criticism or debate is permitted. Don't want to inhibit participants.
- Participants understand nothing they say will be held against them later on.
- Scribe write down all ideas where all can see
e.g., whiteboard, paper taped to wall
- Wild is good. Feel free to be gloriously wrong.

- *Participants should NOT self-censor or spend too much time wondering if an idea is practical. Just shout it out.*
- *Original list does not get circulated outside of the meeting.*

Brainstorming

Part II — The Calm

- Go over the list. Explain ideas more carefully.
- Categorize into “maybe” and “no” by pre-agreed consensus method.
 - *informal consensus, 50% + 1 vs. “clear majority”, Dutch auction, who has vetoes.*
- Be careful about time.
 - *Meetings (esp. if creative or technical in nature) tend to lose focus after 90 to 120 minutes. Take breaks or reconvene later.*
- Review, consolidate, combine, clarify, expand.
- Rank the list by priority somehow; choose a winner.

Pruning -2

There are several choices of how:

voting with threshold

voting with campaign speeches

blending ideas

Voting with threshold

Each person is allowed to vote up to n times.

Keep those ideas with more than m votes.

Have multiple rounds thereof with smaller n and m .

Voting with campaign speeches

Each person is allowed to vote up to $j < n$ times.

Keep those ideas with at least one vote.

Have someone who did not vote for an idea defend it for the next round.

Have multiple rounds thereof with smaller j .

Blending ideas

Apply acceptance criteria (which tend to be ignored in first step) to ideas.

Rank accepted ideas.

Select top k for voting treatment.

Other Brainstorming Ideas

Brainstorming can be carried out over e-mail.

But a leader is needed to prevent flaming and race conditions.

One Final Point!

With lots of good, outrageous, outlandish ideas, the brainstorm is loads of fun!!

Fun motivates people to do well!!!

What Can Go Wrong in Elicitation and the SRS?

- Unknown requirements

The hardest part of writing specifications involves anticipating all of the possible circumstances that might occur. Users expect software systems to respond correctly to whatever input is presented. Problems occur because there are situations that nobody considers during development, and thus the software does not handle the situation when it occurs.

One of the goals of modelling is to reveal circumstances that have not been considered and reveal areas of the problem that need to be explored more with the customer.

- Known but undiscussed requirements (assumptions)

Assume == “ass” of “u” and “me”

- Discussed but undocumented requirements
- Wrongly documented requirements

These requirements are sometimes sabotage from users who don't want the system to succeed, either because they don't want their routines to change, or because their jobs are threatened by the new system.

Other Techniques

- PIECES
- Social and Organizational Factors
- Ethnographic Analysis
- Joint Application Design
- Names and Norms
- Gause & Weinberg Ideas

The PIECES Approach

- A more structured approach than simple brainstorming; think of as a vanilla RE process.
- Works best with existing system or well-understood domain, but perhaps inexperienced requirements engineers.

- Main idea:

→ Examine system from six specified points of view. Provides a lowest common denominator starting point when you are not sure how to get started.

- Oriented towards office information systems (esp. enhancing/modifying existing systems), but concepts are broadly applicable.

- PIECES == **P**erformance, **i**nformation and data, **e**conomy, **c**ontrol, **e**fficiency, and **s**ervices.

- There is overlap between areas, but that's OK; you're examining different points of view.

JAD — Joint Application Design

- Developed at IBM in the 1970s; lots of success stories.
- Think of as “structured brainstorming”, IBM-style. *full of structure, defined roles, forms to be filled out, TLAs*
- Two major “steps”, three phases each, and six (human) roles to be played!
- Four main tenets of JAD:
 1. Effective use of group dynamics. *facilitated and directed group sessions to get common understanding and universal buy-in*
 2. Use of visual aids. *to enhance understanding, e.g., props, prepared diagrams*
 3. Defined process. *i.e., not a random hodgepodge*
 4. Standardized forms for documenting results. *LCD approach*

Other Elicitation Concepts

Here are some twists of G&W concepts that G&W did not think of, but I thought of when reading G&W.

They concern:

- **norms**
- **mockups & prototypes**
- **existence assumption**
- **right-brain methods**
- **naming**

Norms -1

The general form of the use of a norm to state requirements:

Here is an X ; build a better X

The norm can protect you from colossal blunders by starting with something that is clearly feasible.

But, it can keep you from seeing a new way to solve the problem that X , itself, is solving by keeping you immersed in enhancing X .

Norms -2

Here's an example of such a norm:

“Build a better pencil-and-paper set” could prevent you from thinking of the computer as an authoring tool.

Norms -3

Another example:

Avocado is a fruit.

Problem: peel avocado.

Thinking of the norm of fruit causes you to try to peel avocado with knife.

Wotta mess!

Norms -4

A better solution is to scoop the avocado meat out of the peel-shell with a spoon just the right size.

Existence assumption -1

Underlying the whole search for a solution is the assumption that a solution exists.

Generally this assumption is tacitly accepted.

Usually this is just fine.

Existence assumption -2

People have a good sense of when a problem is solvable and when it is not.

But, sometimes it is necessary to examine the existence assumption and verify that it is reasonable.

If the assumption is not reasonable, then maybe the problem has to be changed.

Existence assumption -3

Example:

Situation: I bought rim-mounted woks and used them for years over a gas stove.

Then I moved to an electric stove.

The woks never got hot enough to make the food right; the food came out greasy.

The first attempt to state the problem: Make an electric burner hot enough to heat a rim-mounted wok.

Existence assumption -4

A solution to this problem does not exist, because so hot a burner will damage itself and the rim.

Finally, I realized that the solution was to change the shape of the wok so that the cooking surface is completely on the burner.

So I went out and bought a flat-bottomed wok!

Right-Brain Methods -1

G&W discuss a number of right-brain methods to help overcome communicational ambiguities.

The left brain is the more textual, logical half.

The right brain is the more pictorial, free associating half.

Right-Brain Methods -2

When you do not understand someone, ask him or her to draw a diagram showing his or her meaning.

Or, draw your conception of what he or she is saying and ask if this is what he or she means

This is sort of like what was done using the holodeck on *Star Trek, the Next Generation* episode “Schisms”.

Naming -1

What's in a name? A rose by any other name would smell as sweet!

Ah, but if the rose were not visible or smellable and someone asked you if you wanted a vered would you answer “yes” as quickly as you might if you were asked if you wanted a rose?

“Vered” is Hebrew for “rose”.

Naming -2

What if you were asked if you wanted a qwiddlyhop?

G&W show how a bad name can distract a project and how a good name can be an inspiration to all that work with it.

Naming -3

G&W discuss how an inaccurate name can mislead those who perceive it and cause clashes when confronting the real thing.

So, it is worth taking time out to brainstorm for a good name.

Naming -4

Be careful with backcronyms

A backronym is clever name that is made after the fact, an acronym for a contrived sequence of words.

Those words may not accurately describe the project, and may eventually mislead newcomers, clients, and users.

Naming -5

Getting the right name is like getting the right norm.

“Post-its” suggests better uses than does “half-sticky adhesive”.

“Scoop out meat of avocado” suggests a better solution than “peel off skin of avocado”.

Words

A picture is worth a thousand words.

One word is worth a thousand pictures.

That word is an abstraction of the thousand pictures, and of the million words that are worth the pictures.

Expectations -4

The client, just as anyone else, has expectations.

The difference 'twixt disappointment and delight over a product is how well expectations are matched upon delivery of the product.

Expectations -5

Sometimes, a client's expectations are too high.

Perhaps, the client has developed unreasonable expectations for the product from having seen other products or movies.

Expectations -6

Perhaps, the client has not read the fine print.

Actually, there should be *no* fine print; if there is, you, the requirements engineer, have not done your job.

It is your job to limit the client's expectations to something reasonable.

Expectations -7

Reasons to limit expectations:

If all prior steps were done perfectly, expectation limitation would be redundant, but

- **people are not perfect,**
- **people are not logical,**
- **people perceive things differently,**
- **designers are people too.**