Requirements Elicitation for Software

MUSADAQ MANSOOR

Before requirements can be analyzed, modeled, or specified they must be gathered through an elicitation process.

Initiating the Process

- The most commonly used requirements elicitation technique is to conduct a meeting or interview.
- The first meeting between a software engineer (the analyst) and the customer can be likened to the awkwardness of a first date between two strangers.

Recommendations

- ❑Start by asking context free questions
- ❑Who is behind the request for this work? – Who will use the solution? – What will be the economic benefit of a successful solution? – Is there another source for the solution that you need?
- ❑These questions help to identify all stakeholders who will have interest in the software to be built.

Cont....

The next set of questions enables the analyst to gain a better understanding of the problem and the customer to voice his or her perceptions about a solution:

– How would you characterize "good" output that would be

generated by a successful solution?

– What problem(s) will this solution address?

– Can you show me (or describe) the environment in which the

solution will be used?

– Will special performance issues or constraints affect the way

the solution is approached?

Cont....

The final set of questions focuses on the effectiveness of the meeting.

– Are you the right person to answer these questions? Are your answers "official"?

– Are my questions relevant to the problem that you have?

– Am I asking too many questions?

– Can anyone else provide additional information?

– Should I be asking you anything else?

Cont....

These questions (and others) will help to "break the ice" and initiate the communication.

2. Facilitated Application Specification Techniques (FAST)

- ❑Too often, customers and software engineers have an unconscious "us and them" mind-set.
- ❑ Rather than working as a team to identify and refine requirements, each constituency defines its own "territory" and communicates through a series of memos, formal position papers, documents, and question and answer sessions.
- ❑History has shown that this approach doesn't work very well. Misunderstandings flourish, important information is omitted, and a successful working relationship is never established.

FAST

- ❑This approach encourages the creation of a joint team of customers and developers who work together to:
- ❑Identify the problem
- ❑Propose elements of the solution
- ❑Negotiate different approaches and specify a preliminary set of solution requirements.

FAST : Basic guidelines

- – A meeting is conducted at a neutral site and attended by both software engineers and customers.
- – Rules for preparation and participation are established.
- – An agenda is suggested that is formal enough to cover all
- important points but informal enough to encourage the free flow of ideas.
- – A 'facilitator' (can be a; customer, a developer, or an outsider) controls the meeting.
- – A "definition mechanism" (can be work sheets, flip charts, or wall stickers or an electronic bulletin board, chat room or virtual forum) is used.

3. Quality Function Deployment

A quality management technique that translates needs of customers into technical requirements of software.

- ❑**Normal Requirement**: meeting objectives & goals stated for a product or system during meeting
- ❑**Expected Requirement:** Implicit to products / system and may be so fundamental that customer does not explicitly state them
- ❑**Exciting Requirement:** Features beyond customer’s expectation and prove to be very satisfying when present

4. Use Cases

As requirements are gathered as part of:

- Informal meeting
- FAST or QFD

SW Engineer can create a set of scenario that identify a thread of usage for system to be constructed; providing a description of how system will be used.

5. Analysis Principles

- ❑A variety of modeling notations are developed by investigators. Each analysis method has a unique point of view. However all analysis methods are related by a set of operational principles like:
- ❑ The information domain of a problem must be represented and understood.
- ❑The functions that the software is to perform must be defined.
- ❑The behavior of the software (as a sequence of external events) must be represented.
- ❑The analysis process should move from essential information toward implementation detail.

6. Software Prototyping

- ❑Analysis should be conducted regardless of the SW engineering paradigm. (Various approaches apply)
- ❑In some cases it is possible to apply operational analysis principles and derive a model of SW from which a design can be developed.
- ❑In other situation Requirement Elicitation (FAST, QFD etc) is conducted and a model is built, called Prototype.