CHAPTER 7
Requirements elicitation

Objectives

- Student should understand the importance of requirements elicitation in requirement engineering.
- Student should enhance the list of requirements elicitation techniques and how to use them.
- Students should understand which techniques work well? Why? Which ones do not work so well? Why not?
- Which techniques that should use in agile projects, other projects.

Contents

- Requirements elicitation overview
- Requirements elicitation techniques
- Planning elicitation on your project
- Preparing for elicitation
- Performing elicitation activities
- Following up after elicitation
- Classifying customer input
- Some cautions about elicitation
- Finding missing requirement

Requirements elicitation overview

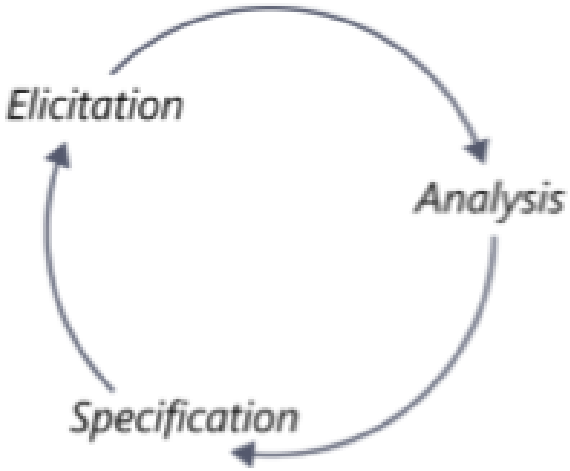

FIGURE 7-1 The cyclic nature of requirements elicitation, analysis, and specification.

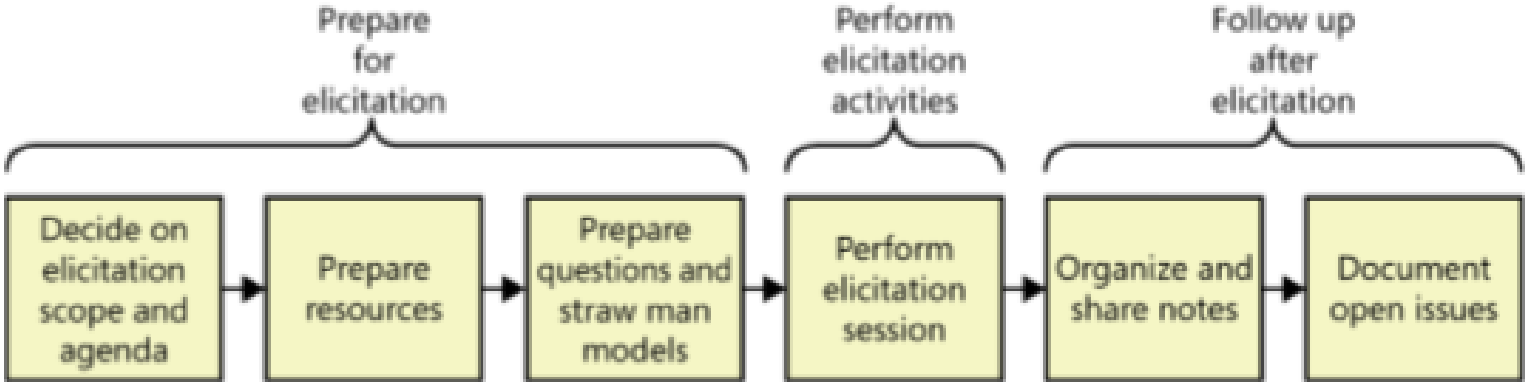

FIGURE 7-2 Activities for a single requirements elicitation session.

Requirements elicitation techniques: Interviews

- Establish rapport
- Stay in scope
- Prepare questions and straw man models ahead of time
- Suggest ideas
- Listen actively

Requirements elicitation techniques: Workshops

- Establish and enforce ground rules
- Fill all of the team roles
- Plan an agenda
- Stay in scope
- Use parking lots to capture items for later consideration
- Timebox discussions
- Keep the team small but include the right stakeholders
- Keep everyone engaged

Requirements elicitation techniques: Focus groups

- A representative group of users who convene in a facilitated elicitation activity to generate input and ideas on a product's functional and quality requirements.
- Focus group sessions must be interactive, allowing all users a chance to voice their thoughts.
- Focus groups are useful for exploring users' attitudes, impressions, preferences, and needs
- Useful for commercial products and don't have ready access to end users within your company.

Requirements elicitation techniques: Observations

- Focus on task elicitation in the system-as-is
- Understanding a task is often easier by observing people performing it (rather than verbal or textual explanation)
  - cf. tying shoelaces
- Passive observation: no interference with task performers
  - Watch from outside, record (notes, video), edit transcripts, interpret
  - Protocol analysis: task performers concurrently explain it
  - Ethnographic studies: over long periods of time, try to discover emergent properties of social group involved
    - about task performance + attitudes, reactions, gestures, ...
- Active observation: you get involved in the task, even become a team member

Requirements elicitation techniques: Questionnaires

- Submit a list of questions to selected stakeholders, each with a list of possible answers (+ brief context if needed)
  - Multiple choice question: one answer to be selected from answer list
  - Weighting question: list of statements to be weighted...
    - qualitatively ('high', 'low", ...), or
    - quantitatively (percentages)
    - to express perceived importance, preference, risk etc.
- Effective for acquiring subjective info quickly, cheaply, remotely from many people
- Helpful for preparing better focussed interviews (see next)

Requirements elicitation techniques: Questionnaires

- Subject to ...
  - multiple biases: recipients, respondents, questions, answers
  - unreliable info: misinterpretation of questions, of answers, inconsistent answers, ...

=> Guidelines for questionnaire design/validation:

  - Select a representative, statistically significant sample of people; provide motivation for responding
  - Check coverage of questions, of possible answers
  - Make sure questions, answers, formulations are unbiased & unambiguous
  - Add implicitly redundant questions to detect inconsistent answers
  - Have your questionnaire checked by a third party

System interface analysis

- Purpose
- What it is?
- How to process
- Practice with student’s assignment

User interface analysis

- Purpose
- What it is?
- How to process
- Practice with student’s assignment

Document analysis

- Purpose
- What it is?
- How to process
- Practice with student’s assignment

Brainstorming

- Purpose
- What it is?
- How to process
- Practice with student’s assignment

Prototyping

- Purpose
- What it is?
- How to process
- Practice with student's assignment

Storyboards

- Purpose
- What it is?
- How to process
- Practice with student’s assignment

Planning elicitation on your project

- Elicitation objectives
- Elicitation strategy and planned techniques
- Schedule and resource estimates
- Documents and systems needed for independent elicitation
- Expected products of elicitation efforts
- Elicitation risks

Preparing for elicitation

- Plan session scope and agenda
- Prepare resources
- Learn about the stakeholders
- Prepare questions
- Prepare straw man models

Performing elicitation activities

- Educate stakeholders
- Take good notes
- Exploit the physical space

Following up after elicitation

- Organizing and sharing the notes
- Documenting open issues

Classifying customer input

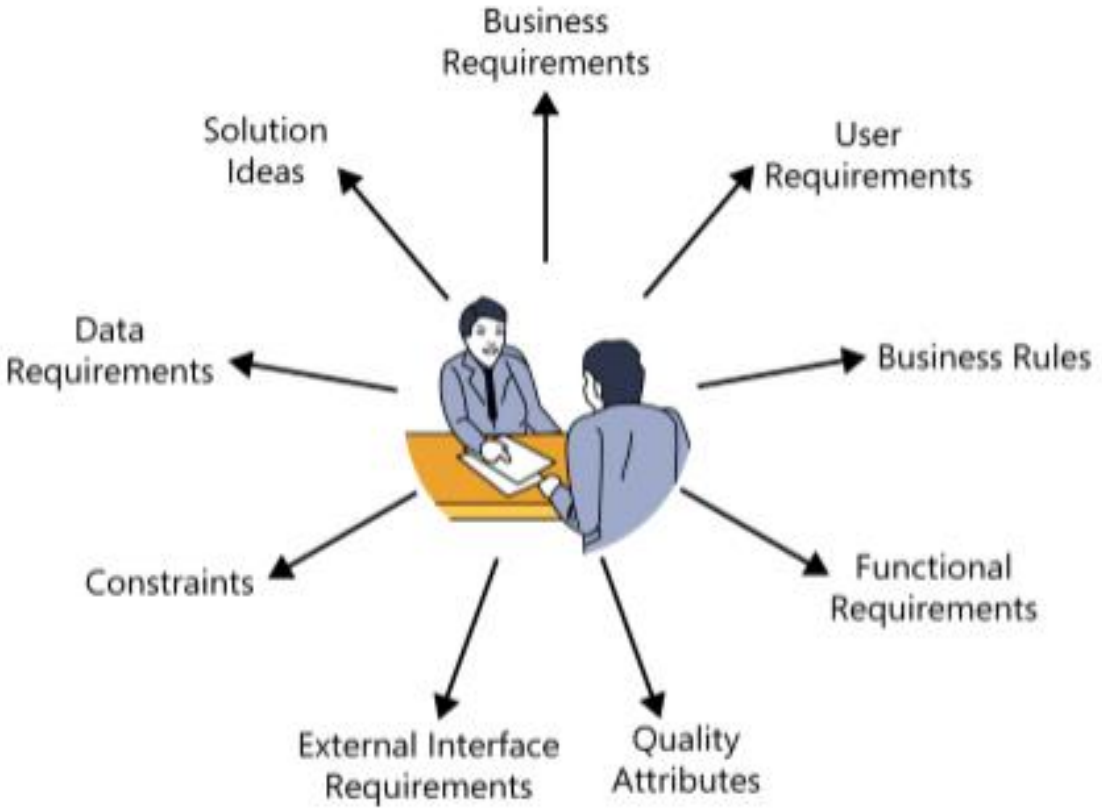

FIGURE 7-7 Classifying customer input.

How do you know when you're done?

- No simple signal will indicate when you've completed requirements elicitation.
- Perhaps you are done if:
  - The users can't think of any more use cases or user stories. Users tend to identify user requirements in sequence of decreasing importance.
  - Users propose new scenarios, but they don't lead to any new functional requirements. A "new" use case might really be an alternative flow for a use case you've already captured.
  - Users repeat issues they already covered in previous discussions.

How do you know when you're done?

- Suggested new features, user requirements, or functional requirements are all deemed to be out of scope.
- Proposed new requirements are all low priority.
- The users are proposing capabilities that might be included "sometime in the lifetime of the product" rather than "in the specific product we're talking about right now."
- Developers and testers who review the requirements for an area raise few questions

Some cautions about elicitation

- Balance stakeholder representation
- Define scope appropriately
- Avoid the requirements-versus-design argument
- Research within reason

Assumed and implied requirements

- Assumed requirements are those that people expect without having explicitly expressed them. What you assume as being obvious might not be the same as assumptions that various developers make.
- Implied requirements are necessary because of another requirement but aren't explicitly stated. Developers can't implement functionality they don't know about.

Finding missing requirements

- Decompose high-level requirements into enough detail to reveal exactly what is being requested
- Ensure that all user classes have provided input
- Trace system requirements, user requirements, event-response lists, and business rules to their corresponding functional requirements to make sure that all the necessary functionality was derived.
- Check boundary values for missing requirements
- Represent requirements information in more than one way

Finding missing requirements

- Sets of requirements with complex Boolean logic (ANDs, ORs, and NOTs) often are incomplete
- Create a checklist of common functional areas to consider for your projects
- A data model can reveal missing functionality