Requirements Engineering

Chapter 3 Requirements Elicitation and Analysis

Learning Objective

...to describe the activities in the requirements engineering process which are
concerned with discovering requirements <u>and</u> analyzing requirements for
incompleteness, *inconsistency*, *relevance* and *practicality,* as well as negotiating the
final requirements for the system.

Frederick T Sheldon
Assistant Professor of Computer Science
University of Colorado at Colorado Springs

CS 531 Software Requirements Analysis and Specification
From *Requirements Engineering Processes and Techniques* by G. Kotonya and I. Sommerville 1998
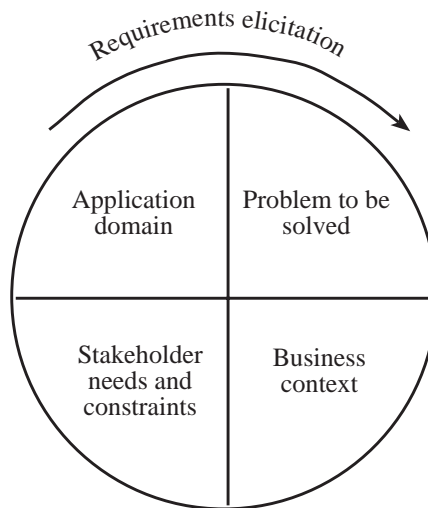
Chapter 3
Slide 1

Objectives

- ⊗ To describe the processes of requirements elicitation and analysis.
- ⊗ To introduce a number of requirements elicitation and requirements analysis techniques.
- ⊗ To discuss how prototypes may be used in the RE process.

CS 531 Software Requirements Analysis and Specification
From *Requirements Engineering Processes and Techniques* by G. Kotonya and I. Sommerville 1998
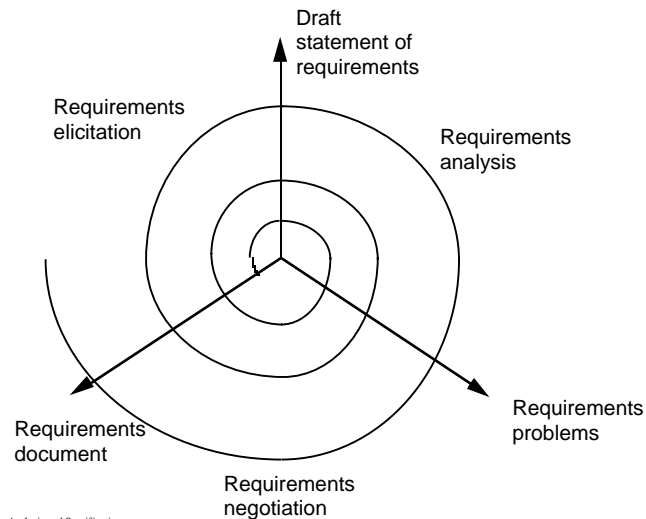
Chapter 3
Slide 2

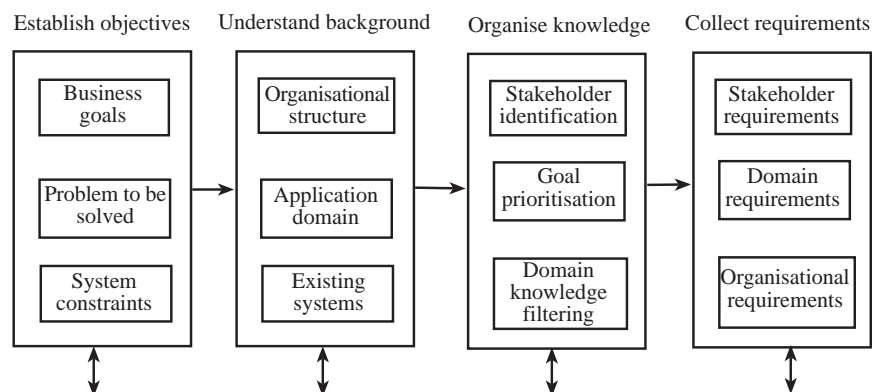# Components of requirements elicitation

Elicitation activities

⊗ **Application domain understanding**
  - Application domain knowledge is knowledge of the general area where the system is applied.

⊗ **Problem understanding**
  - The details of the specific customer problem where the system will be applied must be understood.

⊗ **Business understanding**
  - You must understand how systems interact and contribute to overall business goals.

⊗ **Understanding the needs and constraints of system stakeholders**
  - You must understand, in detail, the specific needs of people who require system support in their work.
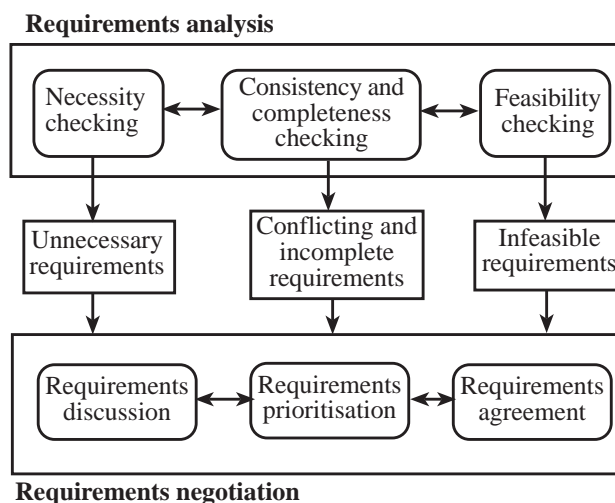
Elicitation, analysis and negotiation

Draft
statement of
requirements

Requirements
elicitation

Requirements
analysis

Requirements
document

Requirements
problems

Requirements
negotiation

The requirements elicitation process

Establish objectives

Understand background

Organise knowledge

Collect requirements

Business goals	Organisational structure	Stakeholder identification	Stakeholder requirements
Problem to be solved	Application domain	Goal prioritisation	Domain requirements
System constraints	Existing systems	Domain knowledge filtering	Organisational requirements

Elicitation stages

⊗ Objective setting
  - The organizational objectives should be established including general goals of the business, an outline description of the problem to be solved, why the system is necessary and the constraints on the system.

⊗ Background knowledge acquisition
  - Background information about the system includes information about the organization where the system is to be installed, the application domain of the system and information about existing systems

⊗ Knowledge organization
  - The large amount of knowledge which has been collected in the previous stage must be organized and collated.

⊗ Stakeholder requirements collection
  - System stakeholders are consulted to discover their requirements.

CS 531 Software Requirements Analysis and Specification
From *Requirements Engineering Processes and Techniques* by G. Kotonya and I. Sommerville 1998

Chapter 3
Slide 7

Requirements analysis and negotiation

Requirements analysis

Necessity checking ↔ Consistency and completeness checking ↔ Feasibility checking

Unnecessary requirements | Conflicting and incomplete requirements | Infeasible requirements

Requirements discussion ↔ Requirements prioritisation ↔ Requirements agreement

Requirements negotiation

CS 531 Software Requirements Analysis and Specification
From *Requirements Engineering Processes and Techniques* by G. Kotonya and I. Sommerville 1998

Chapter 3
Slide 8

Analysis checks

- ⊗ Necessity checking
  - The need for the requirement is analyzed. In some cases, requirements may be proposed which don't contribute to the business goals of the organization or to the specific problem to be addressed by the system.

- ⊗ Consistency and completeness checking
  - The requirements are cross-checked for consistency and completeness. Consistency means that no requirements should be contradictory; completeness means that no services or constraints which are needed have been missed out.

- ⊗ Feasibility checking
  - The requirements are checked to ensure that they are feasible in the context of the budget and schedule available for the system development.

Requirements negotiation

- ⊗ Requirements discussion
  - Requirements which have been highlighted as problematical are discussed and the stakeholders involved present their views about the requirements.

- ⊗ Requirements prioritization
  - Disputed requirements are prioritized to identify critical requirements and to help the decision making process.

- ⊗ Requirements agreement
  - Solutions to the requirements problems are identified and a compromise set of requirements are agreed. Generally, this will involve making changes to some of the requirements.

Elicitation techniques

⊗ Specific techniques which may be used to collect knowledge about system requirements

⊗ This knowledge must be structured
- Partitioning - aggregating related knowledge
- Abstraction - recognizing generalities
- Projection - organizing according to perspective

⊗ Elicitation problems
- Not enough time for elicitation
- Inadequate preparation by engineers
- Stakeholders are unconvinced of the need for a new system

Specific elicitation techniques

⊗ Interviews

⊗ Scenarios

⊗ Soft systems methods

⊗ Observations and social analysis

⊗ Requirements reuse

Interviews

⊗ The requirements engineer or analyst discusses the system with different stakeholders and builds up an understanding of their requirements.

⊗ Types of interview

- Closed interviews. The requirements engineer looks for answers to a pre-defined set of questions
- Open interviews There is no predefined agenda and the requirements engineer discusses, in an open-ended way, what stakeholders want from the system.

Interviewing essentials

⊗ Interviewers must be open-minded and should not approach the interview with pre-conceived notions about what is required

⊗ Stakeholders must be given a starting point for discussion. This can be a question, a requirements proposal or an existing system

⊗ Interviewers must be aware of organizational politics - many real requirements may not be discussed because of their political implications

Scenarios

⊗ Scenarios are stories which explain how a system might be used. They should include

  • a description of the system state before entering the scenario
  • the normal flow of events in the scenario
  • exceptions to the normal flow of events
  • information about concurrent activities
  • a description of the system state at the end of the scenario

⊗ Scenarios are examples of interaction sessions which describe how a user interacts with a system

⊗ Discovering scenarios exposes possible system interactions and reveals system facilities which may be required

---

Library scenario - document ordering

⊗ Log on to EDDIS system

⊗ Issue order document command

⊗ Enter reference number of the required document

⊗ Select a delivery option

⊗ Log out from EDDIS

⊗ This sequence of events can be illustrated in a diagram

Library Scenario
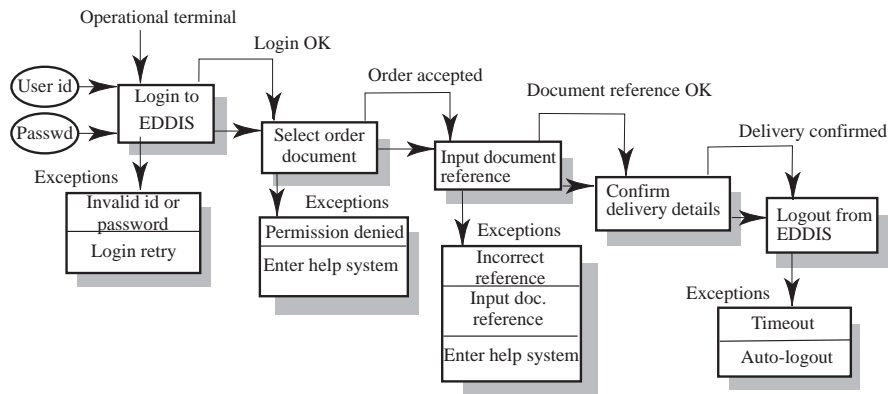
Scenarios and OOD

⊗ Scenarios are an inherent part of some object-oriented development methods

⊗ The term use-case (i.e. a specific case of system usage) is sometimes used to refer to a scenario

⊗ There are different views on the relationship between use-cases and scenarios:

  • A use-case is a scenario
  • A scenario is a collection of use-cases. Therefore, each exceptional interaction is represented as a separate use-case

Soft Systems methods

- ⊗ These produce informal models of a social-technical system. They consider the system, the people and the organization

- ⊗ Not techniques for detailed requirements elicitation. Rather, they are ways of understanding a problem and its organizational context

- ⊗ Software Systems Methodology (SSM) is probably the best known of these methods

- ⊗ The essence of SSM is its recognition that systems are embedded in a wider human and organizational context

CS 531 Software Requirements Analysis and Specification
From *Requirements Engineering Processes and Techniques* by G. Kotonya and I. Sommerville 1998

Chapter 3
Slide 19

Stages of SSM

- ⊗ Problem situation assessment
- ⊗ Problem situation description
- ⊗ Abstract system definition from selected viewpoints
- ⊗ Conceptual system modeling
- ⊗ Model/real-world comparison
- ⊗ Change identification
- ⊗ Recommendations for action

CS 531 Software Requirements Analysis and Specification
From *Requirements Engineering Processes and Techniques* by G. Kotonya and I. Sommerville 1998

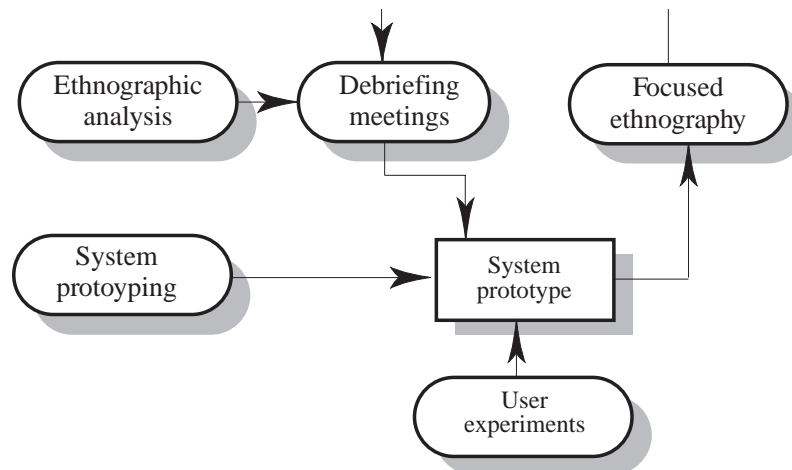Chapter 3
Slide 20

Observation and social analysis

- ⊗ People often find it hard to describe what they do because it is so natural to them. Sometimes, the best way to understand it is to observe them at work

- ⊗ Ethnography is a technique from the social sciences which has proved to be valuable in understanding actual work processes

- ⊗ Actual work processes often differ from formal, prescribed processes

- ⊗ An ethnographer spends some time observing people at work and building up a picture of how work is done

CS 531 Software Requirements Analysis and Specification
From *Requirements Engineering Processes and Techniques* by G. Kotonya and I. Sommerville 1998

Chapter 3
Slide 21

Ethnography guidelines

- ⊗ Assume that people are good at doing their job and look for non-standard ways of working

- ⊗ Spend time getting to know the people and establish a trust relationship

- ⊗ Keep detailed notes of all work practices. Analyze them and draw conclusions from them

- ⊗ Combine observation with open-ended interviewing

- ⊗ Organize regular de-briefing session where the ethnographer talks with people outside the process

- ⊗ Combine ethnography with other elicitation techniques

CS 531 Software Requirements Analysis and Specification
From *Requirements Engineering Processes and Techniques* by G. Kotonya and I. Sommerville 1998

Chapter 3
Slide 22

Ethnography in elicitation

Ethnographic perspectives

⊗ **The work setting viewpoint**
  • This describes the context and the physical location of the work and how people use objects to carry out tasks. Therefore, in a study of a help desk (say), this would describe the objects which the helper had to hand and how these were organized.

⊗ **Social and organizational perspectives**
  • This tries to bring out the day-to-day experience of work as seen by different people who are involved. Each individual typically sees the work in a different ways and this viewpoint tries to organize and integrate all of these perceptions.

⊗ **The workflow viewpoint**
  • This viewpoint presents the work from a series of work activities with information flowing from one activity to another.

Requirements reuse

⊗ Reuse involves taking the requirements which have been developed for one system and using them in a different system

⊗ Requirements reuse saves time and effort as reused requirements have already been analyzed and validated in other systems

⊗ Currently, requirements reuse is an informal process but more systematic reuse could lead to larger cost savings

Reuse possibilities

⊗ Where the requirement is concerned with providing application domain information.

⊗ Where the requirement is concerned with the style of information presentation. Reuse leads to a consistency of style across applications.

⊗ Where the requirement reflects company policies such as security policies.

Prototyping

⊗ A prototype is an initial version of a system which may be used for experimentation

⊗ Prototypes are valuable for requirements elicitation because users can experiment with the system and point out its strengths and weaknesses. They have something concrete to criticize

⊗ Rapid development of prototypes is essential so that they are available early in the elicitation process

Prototyping benefits

⊗ The prototype allows users to experiment and discover what they really need to support their work

⊗ Establishes feasibility and usefulness before high development costs are incurred

⊗ Essential for developing the 'look and feel' of a user interface

⊗ Can be used for system testing and the development of documentation

⊗ Forces a detailed study of the requirements which reveals inconsistencies and omissions

Types of prototyping

⊗ Throw-away prototyping
- intended to help elicit and develop the system requirements.
- The requirements which should be prototyped are those which cause most difficulties to customers and which are the hardest to understand. Requirements which are well-understood need not be implemented by the prototype.

⊗ Evolutionary prototyping
- intended to deliver a workable system quickly to the customer.
- Therefore, the requirements which should be supported by the initial versions of this prototype are those which are well-understood and which can deliver useful end-user functionality. It is only after extensive use that poorly understood requirements should be implemented.

Prototyping costs and problems

⊗ Training costs - prototype development may require the use of special purpose tools

⊗ Development costs - depend on the type of prototype being developed

⊗ Extended development schedules - developing a prototype may extend the schedule although the prototyping time may be recovered because rework is avoided

⊗ Incompleteness - it may not be possible to prototype critical system requirements

Approaches to prototyping

⊗ Paper prototyping
  - a paper mock-up of the system is developed and used for system experiments

⊗ 'Wizard of Oz' prototyping
  - a person simulates the responses of the system in response to some user inputs

⊗ Executable prototyping
  - a fourth generation language or other rapid development environment is used to develop an executable prototype

Executable prototype development

⊗ Fourth generation languages based around database systems

⊗ Visual programming languages such as Visual Basic or ObjectWorks

⊗ Internet-based prototyping solutions based on World Wide Web browsers and languages such as Java

Requirements analysis

⊗ The goal of analysis is to discover problems, incompleteness and inconsistencies in the elicited requirements. These are then fed back to the stakeholders to resolve them through the negotiation process

⊗ Analysis is interleaved with elicitation as problems are discovered when the requirements are elicited

⊗ A problem checklist may be used to support analysis. Each requirement may be assessed against the checklist

Analysis checklists

⊗ Premature design
  • Does the requirement include premature design or implementation information?

⊗ Combined requirements
  • Does the description of a requirement describe a single requirement or could it be broken down into several different requirements?

⊗ Unnecessary requirements
  • Is the requirement 'gold plating'? That is, is the requirement a cosmetic addition to the system which is not really necessary.

⊗ Use of non-standard hardware
  • Does the requirement mean that non-standard hardware or software must be used? To make this decision, you need to know the computer platform requirements.

Analysis checklists

- ⊗ Conformance with business goals
  - Is the requirement consistent with the business goals defined in the introduction to the requirements document? Requirements ambiguity
- ⊗ Requirements ambiguity
  - Is the requirement ambiguous i.e. could it be read in different ways by different people? What are the possible interpretations of the requirement?
- ⊗ Requirements realism
  - Is the requirement realistic given the technology which will be used to implement the system?
- ⊗ Requirements testability
  - Is the requirement testable, that is, is it stated in such a way that test engineers can derive a test which can show if the system meets that requirement?

Requirements interactions

- ⊗ A very important objective of requirements analysis is to discover the interactions between requirements and to highlight requirements conflicts and overlaps
- ⊗ A requirements interaction matrix shows how requirements interact with each other. Requirements are listed along the rows and columns of the matrix
  - For requirements which conflict, fill in a 1
  - For requirements which overlap, fill in a 1000
  - For requirements which are independent, fill in a 0

Interaction matrices

Requirement	R1	R2	R3	R4	R5	R6
R1	0	0	1000	0	1	1
R2	0	0	0	0	0	0
R3	1000	0	0	1000	0	1000
R4	0	0	1000	0	1	1
R5	1	0	0	1	0	0
R6	1	0	1000	1	0	0

Requirements negotiation

⊗ Disagreements about requirements are inevitable when a system has many stakeholders. Conflicts are not 'failures' but reflect different stakeholder needs and priorities

⊗ Requirements negotiation is the process of discussing requirements conflicts and reaching a compromise that all stakeholders can agree to

⊗ In planning a requirements engineering process, it is important to leave enough time for negotiation. Finding an acceptable compromise can be time-consuming

Negotiation meetings

⊗ An information stage where the nature of the problems associated with a requirement is explained.

⊗ A discussion stage where the stakeholders involved discuss how these problems might be resolved.

- All stakeholders with an interest in the requirement should be given the opportunity to comment. Priorities may be assigned to requirements at this stage.

⊗ A resolution stage where actions concerning the requirement are agreed.

- These actions might be to delete the requirement, to suggest specific modifications to the requirement or to elicit further information about the requirement.

Key points

⊗ Requirements elicitation involves understanding the application domain, the specific problem to be solved, the organizational needs and constraints and the specific facilities needed by system stakeholders.

⊗ The processes of requirements elicitation, analysis and negotiation are iterative, interleaved processes which must usually be repeated several times.

⊗ There are various techniques of requirements elicitation which may be used including interviewing, scenarios, soft systems methods, prototyping and participant observation.

Key points

⊗ Prototypes are effective for requirements elicitation because stakeholders have something which they can experiment with to find their real requirements.

⊗ Checklists are particularly useful as a way of organizing the requirements validation process. They remind analysts what to look for when reading through the proposed requirements.

⊗ Requirements negotiation is always necessary to resolve requirements conflicts and remove requirements overlaps. Negotiation involves information interchange, discussion and resolution of disagreements.

CS 531 Software Requirements Analysis and Specification
From *Requirements Engineering Processes and Techniques* by G. Kotonya and I. Sommerville 1998

Chapter 3
Slide 41