Requirements Elicitation and Specification

Idaho State University | Computer Science

Isaac Griffith

CS 3321
Department of Computer Science
Idaho State University

ROAR

Topics covered

- Requirements elicitation
- Requirements specification

ROAR

Requirements elicitation and analysis

- Sometimes called requirements elicitation or requirements discovery.

- Involves technical staff working with customers to find out about the application domain, the services that the system should provide and the system's operational constraints.

- May involve end-users, managers, engineers involved in maintenance, domain experts, trade unions, etc. These are called stakeholders.

ROAR

Requirements elicitation

- Software engineers work with a range of system stakeholders to find out about the application domain, the services that the system should provide, the required system performance, hardware constraints, other systems, etc.
- Stages include:
  - Requirements discovery,
  - Requirements classification and organization,
  - Requirements prioritization and negotiation,
  - Requirements specification

Problems of requirements elicitation

- Stakeholders don't know what they really want
- Stakeholders express requirements in their own terms.
- Different stakeholders may have conflicting requirements.
- Organizational and political factors may influence the system requirements.
- The requirements change during the analysis process. New stakeholders may emerge and the business environment may change.

ROAR

Req'ts elicitation and analysis
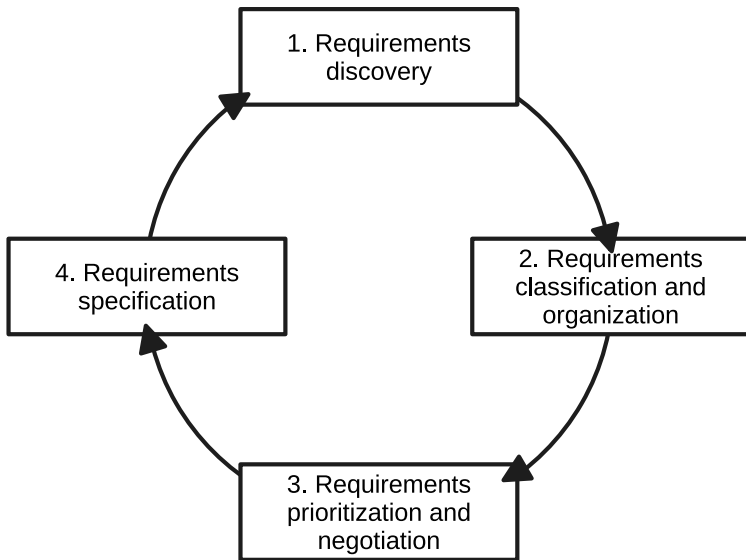
Process activities

- Requirements discovery
  - Interacting with stakeholders to discover their requirements. Domain requirements are also discovered at this stage.

- Requirements classification and organization
  - Groups related requirements and organizes them into coherent clusters.

- Prioritization and negotiation
  - Prioritizing requirements and resolving requirements conflicts

- Requirements specification
  - Requirements are documented and input into the next round of the spiral

ROAR

Requirements Discovery

- The process of gathering information about the required and existing systems and distilling the user and system requirements from this information

- Interaction is with system stakeholders from managers to external regulators

- Systems normally have a range of stakeholders

Interviewing

- Formal or informal interviews with stakeholders are part of most RE processes.
- Types of interview
  - Closed interviews based on pre-determined list of questions
  - Open interviews where various issues are explored with stakeholders
- Effective interviewing
  - Be open-minded, avoid pre-conceived ideas about the requirements and are willing to listen to stakeholders.
  - Prompt the interviewee to get discussions going using a springboard question, a requirements proposal, or by working together on a prototype system.

ROAR

Interviews in practice

- Normally a mix of closed and open-ended interviewing

- Interviews are good for getting an overall understanding of what stakeholders do and how they might interact with the system.

- Interviewers need to be open-minded without pre-conceived ideas of what the system should do.

- You need to prompt the user to talk about the system by suggesting requirements rather than simply asking them what they want.

ROAR

- Application specialists may use language to describe their work that isn't easy for the requirements engineer to understand

- Interviews are not good for understanding domain requirements
  - Requirements engineers cannot understand specific domain terminology;
  - Some domain knowledge is so familiar that people find it hard to articulate or think that it isn't worth articulating

Ethnography

- A social scientist spends a considerable time observing and analyzing how people actually work.

- People do not have to explain or articulate their work.

- Social and organizational factors of importance may be observed.

- Ethnographic studies have shown that work is usually richer and more complex than suggested by simple system models.

ROAR

Scope of ethnography

- Requirements that are derived from the way that people actually work rather than the way in which process definitions suggest that they ought to work.

- Requirements that are derived form cooperation and awareness of other people's activities
  - Awareness of what other people are doing leads to changes in the ways in which we do things.

- Ethnography is effective for understanding existing processes but cannot identify new features that should be added to a system.

Focused ethnography

- Developed in a project studying the air traffic control process
- Combines ethnography with prototyping
- Prototype development results in unanswered questions which focus the ethnographic analysis.
- The problem with ethnography is that it studies existing practices which may have some historical basis which is no longer relevant.

ROAR
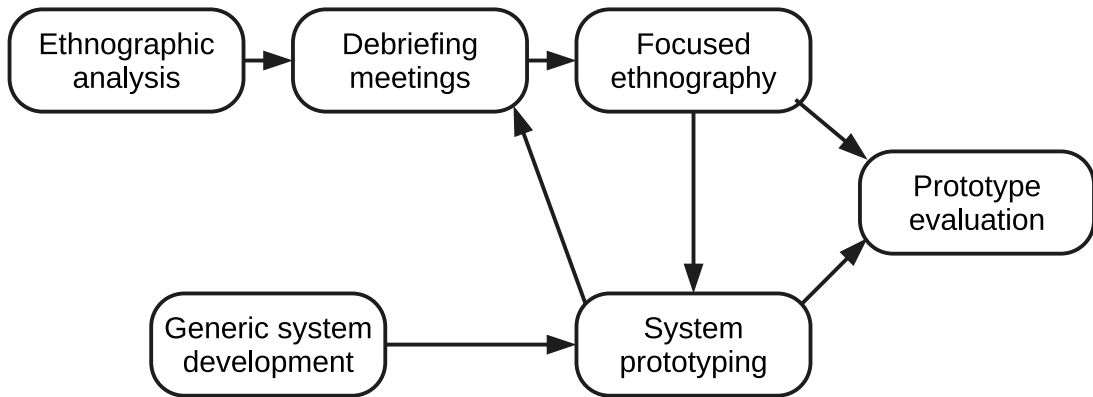
Stories and Scenarios

- Scenarios and user stories are real-life examples of how a system can be used

- Stories and scenarios are a description of how a system may be used for a particular task.

- Because they are based on a practical situation, stakeholders can relate to them and can comment on their situation with respect to the story.

Photo sharing in the classromm

Jack is a primary school teacher in Ullapool (a village in northern Scotland). He has decided that a class project should be focused around the fishing industry in the area, looking at the history, development and economic impact of fishing. As part of this, pupils are asked to gather and share reminiscences from relatives, use newspaper archives and collect old photographs related to fishing and fishing communities in the area. Pupils use an iLearn wiki to gather together fishing stories and SCRAN (a history resources site) to access newspaper archives and photographs. However, Jack also needs a photo sharing site as he wants pupils to take and comment on each others' photos and to upload scans of old photographs that they may have in their families.

Jack sends an email to a primary school teachers group, which he is a member of to see if anyone can recommend an appropriate system. Two teachers reply and both suggest that he uses KidsTakePics, a photo sharing site that allows teachers to check and moderate content. As KidsTakePics is not integrated with the iLearn authentication service, he sets up a teacher and a class account. He uses the iLearn setup service to add KidsTakePics to the services seen by the pupils in his class so that when they log in, they can immediately use the system to upload photos from their mobile devices and class computers.

ROAR

Scenarios

- A structured form of user story
- Scenarios should include
  - A description of the starting situation;
  - A description of the normal flow of event;
  - A description of what can go wrong;
  - Information about other concurrent activities;
  - A description of the state when the scenario finishes.

Uploading Photos (iLearn)

- **Initial assumption**: A user or a group of users have one or more digital photographs to be uploaded to the picture sharing site. These are saved on either a tablet or laptop computer. They have successfully logged on to KidsTakePics.

- **Normal**: The user chooses upload photos and they are prompted to select the photos to be uploaded on their computer and to select the project name under which the photos will be stored. They should also be given the option of inputting keywords that should be associated with each uploaded photo. Uploaded photos are named by creating a conjunction of the user name with the filename of the photo on the local computer.

- On completion of the upload, the system automatically sends an email to the project moderator asking them to check new content and generates an on-screen message to the user that this has been done.

ROAR

Uploading Photos

- **What can go wrong:**
  - No moderator is associated with the selected project. An email is automatically generated to the school administrator asking them to nominate a project moderator. Users should be informed that there could be a delay in making their photos visible.
  - Photos with the same name have already been uploaded by the same user. The user should be asked if they wish to re-upload the photos with the same name, rename the photos or cancel the upload. If they chose to re-upload the photos, the originals are overwritten. If they chose to rename the photos, a new name is automatically generated by adding a number to the existing file name.

- **Other activities**: The moderator may be logged on to the system and may approve photos as they are uploaded.

- **System state on completion**: User is logged on. The selected photos have been uploaded and assigned a status 'awaiting moderation'. Photos are visible to the moderator and to the user who uploaded them.

ROAR

Requirements specification

Requirements specification

- The process of writing down the user and system requirements in a requirements document

- User requirements have to be understandable by end-users and customers who do not have a technical background.

- System requirements are more detailed requirements are more detailed requirements and may include more technical information

- The requirements may be part of a contract for the system development
  - It is therefore important that these are as complete as possible.

ROAR

Writing sys req'ts specification

Notation	Description
Natural language	The requirements are written using numbered sentences in natural language. Each sentence should express one requirement.
Structured natural language	The requirements are written in natural language on a standard form or template. Each field provides information about an aspect of the requirement.
Design description languages	This approach uses a language like a programming language, but with more abstract features to specify the requirements by defining an operational model of the system. This approach is now rarely used although it can be useful for interface specifications.
Graphical notations	Graphical models, supplemented by text annotations, are used to define the functional requirements for the system; UML use case and sequence diagrams are commonly used.
Mathematical specifications	These notations are based on mathematical concepts such as finite-state machines or sets. Although these unambiguous specifications can reduce the ambiguity in a requirements document, most customers don't understand a formal specification. They cannot check that it represents what they want and are reluctant to accept it as a system contract

ROAR

Requirements and design

- In principle, requirements should state what the system should do and the design should describe how it does this.

- In practice, requirements and design are inseparable
  - A system architecture may be designed to structure the requirements;
  - The system may inter-operate with other systems that generate design requirements;
  - The use of a specific architecture to satisfy non-functional requirements may be a domain requirement.
  - This may be the consequence of a regulatory requirement.

Natural language specification

- Requirements are written as natural language sentences supplemented by diagrams and tables

- Used for writing requirements because it is expressive, intuitive and universal. This means that the requirements can be understood by users and customers.

Guidelines for writing req'ts

- Invent a standard format and use it for all requirements.
- Use language in a consistent way. Use **shall** for mandatory requirements, **should** for desirable requirement.
- Use text highlighting to identify key parts of the requirement.
- Avoid the use of computer jargon.
- Include an explanation (rationale) of why a requirement is necessary.

Problems with natural language

- Lack of clarity
  - Precision is difficult without making the document difficult to read.

- Requirements confusion
  - Functional and non-functional requirements tend to be mixed-up

- Requirements amalgamation
  - Several different requirements may be expressed together.

Insulin pump software ex req'ts

3.2 The system shall measure the blood sugar and deliver insulin, if required, every 10 minutes. (Changes in blood sugar are relatively slow so more frequent measurement is unnecessary; less frequent measurement could lead to unnecessarily high sugar levels.)

3.6 The system shall run a self-test routine every minute with the conditions to be tested and the associated actions defined in Table 1. (A self-test routine can discover hardware and software problems and alert the user to the fact the normal operation may be impossible.)

Structured specifications

- An approach to writing requirements where the freedom of the requirements writer is limited and requirements are written in a standard way.

- This works well for some types of requirements e.g., requirements for embedded control system but is sometimes too rigid for writing business system requirement.

ROAR

Form-based specifications

- Definition of the function or entity.
- Description of inputs and where they come from.
- Description of outputs and where they go to.
- Information about the information needed for the computation and other entities used.
- Description of the action to be taken.
- Pre and post conditions (if appropriate)
- The side effects (if any) of the function.

Structured specification example

Insulin Pump/Control Software/SRS/3.3.2

Function Compute insulin dose: safe sugar level.

Description

Computes the dose of insulin to be delivered when the current measured sugar level is in the safe zone between 3 and 7 units.

Inputs Current sugar reading (r2); the previous two readings (r0 and r1)

Source Current sugar reading from sensor. Other readings from memory.

Outputs CompDose—the dose in insulin to be delivered

Destination Main control loop.

Structured specification example

Action

CompDose is zero if the sugar level is stable or falling or if the level is increasing but the rate of increase is decreasing. If the level is increasing and the rate of increase is increasing, then CompDose is computed by dividing the difference between the current sugar level and the previous level by 4 and rounding the results. If the result, is rounded to zero then CompDose is set to the minimum dose that can be delivered.

Requirements

Two previous readings so that the rate of change of sugar level can be computed.

Pre-condition

The insulin resevoir contains at least the maximum allowed single dose of insulin.

Tabular specification

- Used to supplement natural language

- Particularly useful when you have to define a number of possible alternative courses of action

- For example, the insulin pump systems bases its computations on the rate of change of blood sugar level and the tabular specification explains how to calculate the insulin requirement for different scenarios.

Tabular specification example

| Condition	Action
Sugar level falling ($r2 < r1$)	CompDose = 0
Sugar level stable ($r2 = r1$)	CompDose = 0
Sugar level increasing and rate of increase decreasing (($r2 - r1$) < ($r1 - r0$))	CompDose = 0
Sugar level increasing and rate of increase stable or increasing (($r2 - r1$) $\geq$ ($r1 - r0$))	CompDose = round (($r2 - r1$) / 4) if rounded result = 0 then CompDose = MinimumDose

ROAR

Use cases

- Use-cases are a kind of scenario that are included in the UML.

- Use cases identify the actors in an interaction and which describe the interaction itself.

- A set of use cases should describe all possible interactions with the system.

- High-level graphical model supplemented by more detailed tabular description.

- UML sequence diagrams may be used to add detail to use-cases by showing the sequence of event processing in the system.

ROAR

Mentcare system use cases
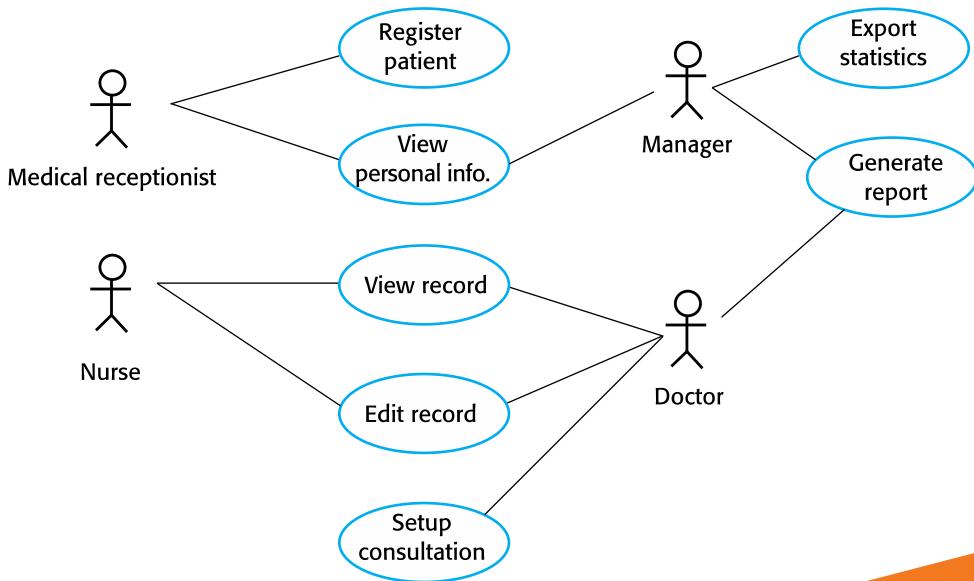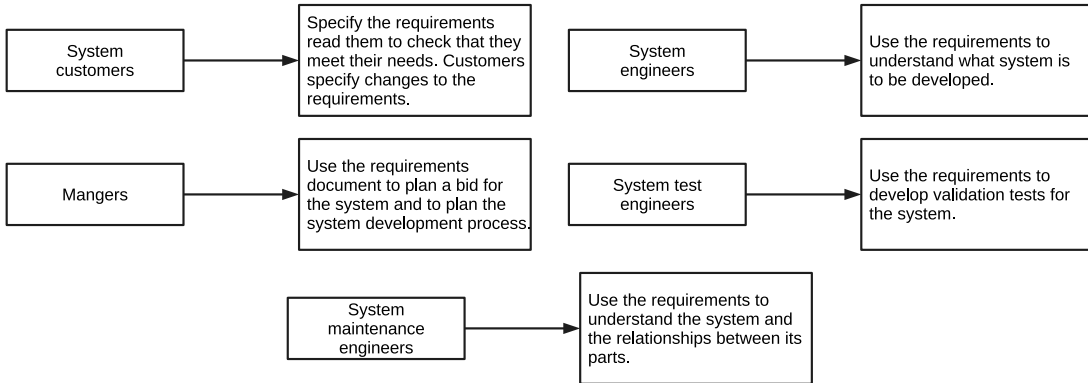
The software requirements document

- The software requirements document is the official statement of what is required of the system developers.

- Should include both a definition of user requirements and a specification of the system requirements.

- It is NOT a design document. As far as possible, it should set WHAT the system should do rather than HOW it should do it.

ROAR

Requirements document users

System customers	→	Specify the requirements read them to check that they meet their needs. Customers specify changes to the requirements.

System engineers	→	Use the requirements to understand what system is to be developed.

Mangers	→	Use the requirements document to plan a bid for the system and to plan the system development process.

System test engineers	→	Use the requirements to develop validation tests for the system.

System maintenance engineers	→	Use the requirements to understand the system and the relationships between its parts.

Requirements document variability

- Information in requirements document depends on type of system and the approach to development used.

- System developed incrementally will, typically, have less detail in the requirements document.

- Requirements documents standards have been designed e.g., IEEE standard. These are mostly applicable to the requirements for large systems engineering projects.

Requirements document structure

Chapter	Description
Preface	This should define the expected readership of the document and describe its version history, including a rationale for the creation of a new version and a summary of the changes made in each version.
Introduction	This should describe the need for the system. It should briefly describe the system's functions and explain how it will work with other systems. It should also describe how the system fits into the overall business or strategic objectives of the organization commissioning the software.
Glossary	This should define the technical terms used in the document. You should not make assumptions about the experience or expertise of the reader.
User requirements definition	Here, you describe the services provided for the user. The nonfunctional system requirements should also be described in this section. This description may use natural language, diagrams, or other notations that are understandable to customers. Product and process standards that must be followed should be specified.

ROAR

Requirements document structure

Chapter	Description
System Architecture	This chapter should present a high-level overview of the anticipated system architecture, showing the distribution of functions across system modules. Architectural components that are reused should be highlighted.
System requirements specification	This should describe the functional and nonfunctional requirements in more detail. If necessary, further detail may also be added to the nonfunctional requirements. Interfaces to other systems may be defined.
System models	This might include graphical system models showing the relationships between the system components and the system and its environment. Examples of possible models are object models, data-flow models, or semantic data models.
System evolution	This should describe the fundamental assumptions on which the system is based, and any anticipated changes due to hardware evolution, changing user needs, and so on. This section is useful for system designers as it may help them avoid design decisions that would constrain likely future changes to the system.

ROAR

Requirements document structure

Chapter	Description
Appendices	These should provide detailed, specific information that is related to the application being developed; for example, hardware and database descriptions. Hardware requirements define the minimal and optimal configurations for the system. Database requirements define the logical organization of the data used by the system and the relationships between data.
Index	Several indexes to the document may be included. As well as a normal alphabetic index, there may be an index of diagrams, an index of functions, and so on.

Key points

- The requirements engineering process is an iterative process that includes requirements elicitation, specification and validation

- Requirements elicitation is an iterative process that can be represented as a spiral of activities – requirements discovery, requirements classification and organization, requirements negotiation and requirements documentation.

- You can use a range of techniques for requirements elicitation including interviews and ethnography. User stories and scenarios may be used to facilitate discussions.

ROAR

Key points

- Requirements specification is the process of formally documenting the user and system requirements and creating a software requirements document.

- The software requirements document is an agreed statement of the system requirements. It should be organized so that both system customers and software developers can use it.

ROAR

Are there any questions?

ROAR