Software Requirements

Descriptions and specifications of a system

Example

……

4.A.5 The database shall support the generation and control of configuration objects; that is, objects which are themselves groupings of other objects in the database. The configuration control facilities shall allow access to the objects in a version group by the use of an incomplete name.

……

Types of requirements

User requirements readers

- Client managers
- System end-users
- Client engineers
- Contractor managers
- System architects

System requirements readers

- System end-users
- Client engineers
- System architects
- Software developersSystem

Software specification readers

- Client engineers (maybe)
- System architects
- Software developers

Examples of functional requirements

Requirements imprecision

- Problems arise when requirements are not precisely stated
- Ambiguous requirements may be interpreted in different ways by developers and users
- Consider the term ‘appropriate viewers’
  - User intention - special purpose viewer for each different document type
  -

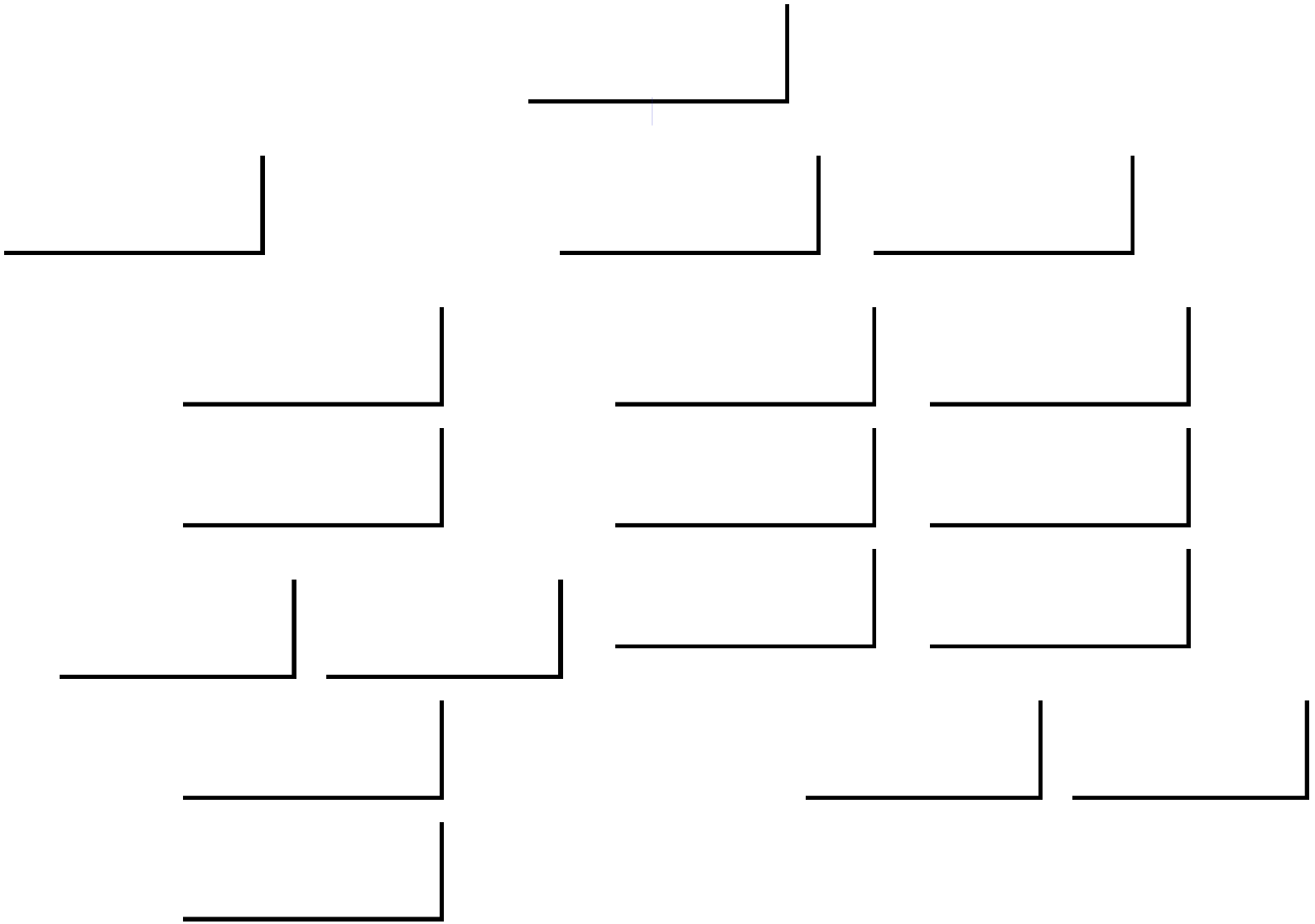
Non-functional requirements

- constraints on the services or functions offered by the system such as timing constraints, constraints on the development process, standards, etc.

Non-functional requirements

- Define system properties and constraints

Non-functional requirement types



Non-functional requirements examples

- Product requirement
  - 4.C.8 It shall be possible for all necessary communication between the APSE and the user to be expressed in the standard Ada character set
- Organizational requirement
  - 9.3.2 The system development process and deliverable documents shall conform to the process and deliverables defined in XYZCo-SP-STAN-95
- External requirement
  - 7.6.5 The system shall not disclose any personal information about customers apart from their name and reference number to the operators of the system

Goals and requirements

- Non-functional requirements may be very difficult tdtate preciselyls animprecise

Examples

- A system goal
  - The system should be easy to use by experienced controllers and should be organized in such a way that user errors are minimized.
- A verifiable non-functional requirement
  - Experienced controllers shall be able to use all the system functions after a total of two hours training. After this training, the average number of errors made by experienced users shall not exceed twoersre

Requirements measures

	•Time to restart after failure
	•PercPePefailure
	•
Portability	•Percentage of target dependent statements
	•Number of target systems

Requirements interaction

- Conflicts between different non-functional requirements are common in complex systems
- Spacecraft system
  - To minimize weight, the number of separate chips in the system should be minimized
  - To minimize power consumption, lower power chips should be used
  - However, using low power chips may mean that more chips have to be used. Which is the most critical requirement?

Domain requirements

- Requirements that come from the application domain of the system and that reflect characteristics of that domain

Domain requirements

- Derived from the application domain and describe system characteristics and features that reflect the domain
- May be new functional requirements, constraints on existing requirements or define specific computations
- If domain requirements are not satisfied, the system may be unworkable

Library system domain requirements

- There shall be a standard user interface to all databases which shall be based on the Z39.50 standard.
- Because of copyright restrictions, some documents must be deleted immediately

Back to user and system requirements

Database requirement

.....

4.A.5 The database shall support the generation and control of

Requirement problems

- Database requirements includes both conceptual and detailed information
  - Describes the concept of configuration control facilities
  - Includes the detail that objects may be accessed using an incomplete name

Editor grid requirement

……

2.6 Grid facilities To assist in the positioning of entities on a diagram, the user may turn on a grid in either centimetres or inches, via an option on the control panel. Initially, the grid is off. The grid may be turned on and off at any time during an editing session and can be toggled between inches and centimetres at any time. A grid option will be provided on the reduce-to-fit view but the number of grid lines shown will be reduced to avoid filling the smaller diagram with grid lines.

……

Requirement problems

- Grid requirement mixes three different kinds of requirement
  - Conceptual functional requirement (the need for a grid)
  - Non-functional requirement (grid units)
  - Non-functional UI requirement (grid switching)

Problems with natural language

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

Detailed user requirement

3.5.1 Adding nodes to a design

3.5.1.1 **The editor shall provide a facility for users to add nodes of a specified type to their design.**

Guidelines for writing requirements

- Invent a standard format and use it for all requirements
- Use language in a consistent way. Use “shall” for mandatory requirements, “should” for desirable requirements
-

System requirements

- More detailed specifications of user requirements
- Serve as a basis for designing the system
- May be used as part of the system contract

Problems with NL specification

- Ambiguity
  - The readers and writers of the requirement must interpret the same words in the same

Alternatives to NL specification

Notation	Description
Structured natural language	This approach depends on defining standard forms or templates to express the requirements specification.
Design description languages	This approach uses a language like a programming language but with more abstr c4(p)-21.2lage ra71.2()-33u.9(r)-4.9purngeoorngboa roai r

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
- Indication of other entities required
- Pre and post conditions (if appropriate)
- The side effects (if any)

Form-based node specification

ECLIPSE/Workstation/Tools/DE/FS/3.5.1

Function Add node

PD

Part of an ATM specification

```
class ATM {
```

PDL disadvantages

- PDL may not be sufficiently expressive to express the system functionality in an understandable way
- Notation is only understandable to people with programming language knowledge
- The requirement may be taken as a design specification rather than a model to help understand the system

Interface specification

- Most systems must operate with other

PDL interface description

```
interface PrintServer {

// defines an abstract printer server
```

The requirements document

- The requirements document is the
  - a
  - NOT

Users of a requirements document

- **System customers**
  - Specify the requirements and read them to check that they meet their needs
- **Managers**
  - Use the requirements document to plan a bid for the system and to plan the system
- **System engineers**
  - Use the requirements to understand what system is to be developed
- **System test engineers**
  - Use the requirements to develop validation tests for the system
- **System maintenance engineers**
  - Use the requirements to help understand the system and the relationship between its parts

Requirements document requirements

- Specify external system behaviour
- Specify implementation constraints
- Easy to change
- Serve as reference tool for maintenance
- Record forethought about the life cycle of the system i.e. predict changes
- Characterise responses to unexpected events

Requirements document

Elicit: by Webster dictionary

Main Entry: **elic·it**
Pronunciation: i-'li-s&t
Function: *transitive verb*
Etymology: Latin elicitus, past participle of elicere, from e- + lacere to allure
Date: 1605
1 : to draw forth or bring out (something latent or potential) <*hypnotism elicited his hidden fears*>
2 : to call forth or draw out (as information or a response) <*her remarks elicited cheers*>

The Requirements Engineering Process

Feasibility studies

- A feasibility study decides whether or not the proposed system is worthwhile
- A short focused study that checks
  - If the system contributes to organizational objectives
  - If the system can be engineered using current technology and within budget
  - If the system can be integrated with other systems that are used

Problems of requirements analysis

- conflicting of r

Banking ATM system

- The example used here is an auto-teller system which provides some automated banking services
- I use a very simplified system which offers some services to customers of the bank who own the system and a narrower range of services to other customers
- Services include cash withdrawal, message passing (send a message to request a service), ordering a statement and transferring funds

Types of viewpoints

Method-based analysis

- Widely used approach to requirements analysis. Depends on the application of a structured method to understand the system
- Methods have different emphases. Some are designed for requirements elicitation, others are close to design methods
- A viewpoint-oriented method (VORD) is used as an example here. It also illustrates the use of viewpoints

The VORD method

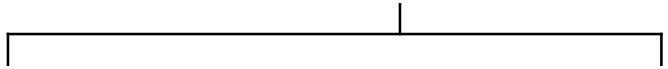
VORD standard forms

Viewp

Viewpoint identification

Viewpoint service information

Viewpoint hierarchy



Customer/cash withdrawal templates

- Reference

Event scenarios

- Event scenarios may be used to describe how a system responds to the occurrence of some particular event such as ‘start transaction’
- VORD includes a diagrammatic convention for event scenarios.
  - Data provided and delivered
  - Control information
  - Exception processing
  - The next expected event

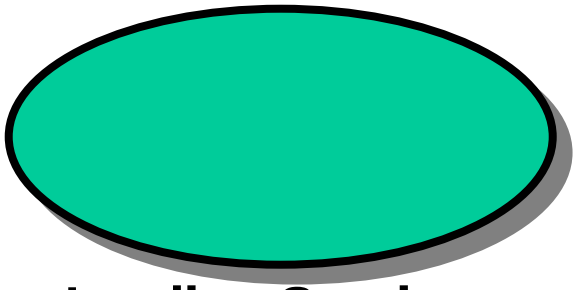

Retain card

Exception description

- Most methods do not include facilities for describing exceptions
- In this example, some exceptions are
  - Timeout. Customer fails to enter a PIN within the allowed time limit
  - Invalid card. The card is not recognized and is returned
  - Stolen card. The card has been registered as stolen and is retained by the machine

Use cases

Lending use-case



Lending Services

Use cases

- Sequence diagrams may be used to add detail to use-cases by showing

Catalogue management: Sequence Diagram

Uncatalog item

Example

- Consider a system that allows senior management to access information without going through middle managers
  - Manageriagtatus. Seni5(oe manageoe y feelor)]TJ 0 -1.

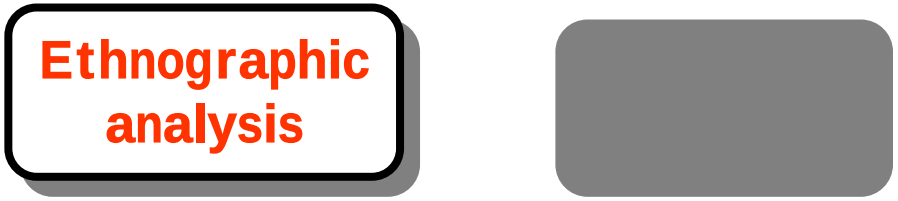
Ethnography

Ethnography

Focused ethnography

- Developed in a project studying the air traffic control process
- Combines ethnography with prototyping
-

Ethnography and prototyping



Scope of ethnography

- Requirements that are derived from the way that people actually work rather than the way process definitions suggest that they ought Require

Requirements checking

- Validity. Does the system provide the functions that best support the customer's needs?
- Consistency. Are there any requirements conflicts?
- Completeness. Are all functions required

Requirements validation techniques

- Requirements reviews
  - Systematic manual analysis of the requirements
- Prototyping
  - Using an executable model of the system to check requirements.
- Test-case generation
  - Developing tests for requirements to check testability
- Automated consistency analysis
  - Checking the consistency of a structured requirements description

Requirements reviews

- Regular reviews should be held while the requirements definition is being formulated
- Both client and contractor staff should be involved in reviews
- Reviews may be formal (with completed documents) or informal. Good communications between developers, customers and users can resolve problems at an early stage

Automated consistency checking

Requirements

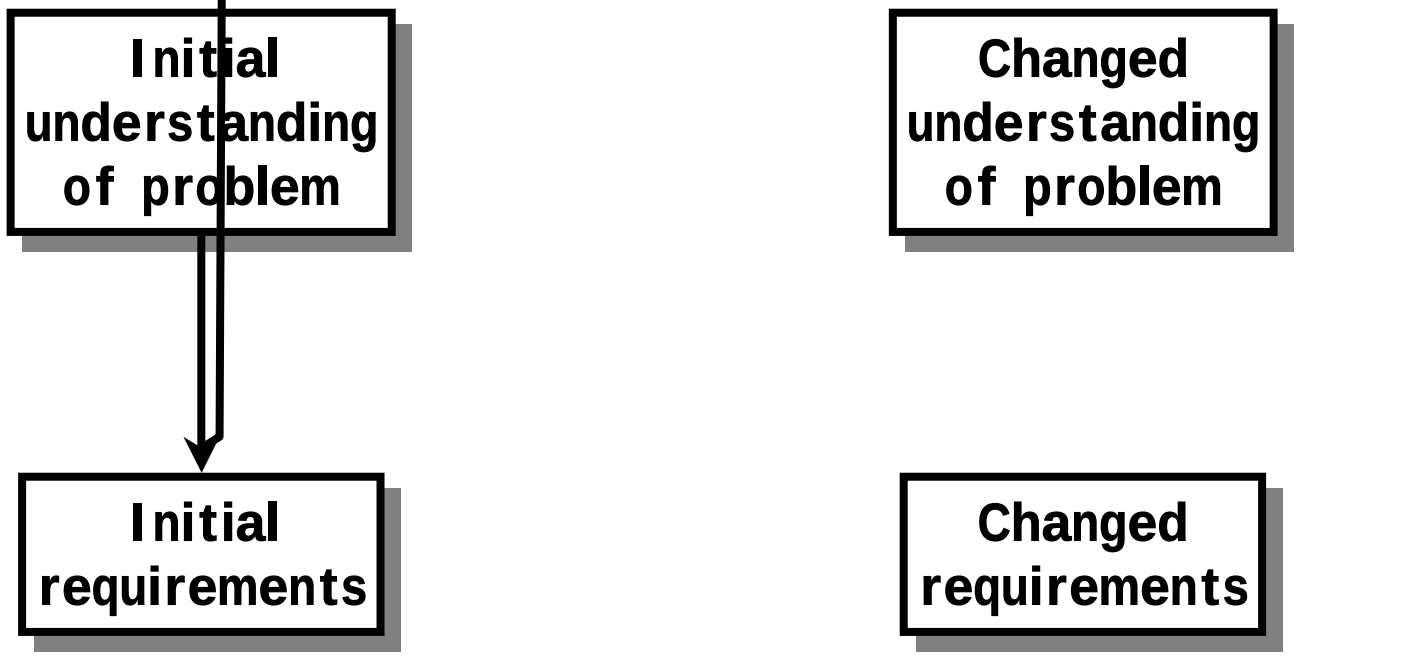
Requirements management

- Requirements management is the process of managing changing requirements during the requirements engineering process and system development
- Requirements are inevitably incomplete ts drsypment•

Requirements change

- The priority of requirements from different viewpoints changes during the development process
- System customers may specify requirements from a business

Requirements evolution



Enduring and volatile requirements

- Enduring requirements. Stable requirements derived from the core activity of the customer organisation. E.g. a hospital will always have doctors, nurses, etc. May be derived from domain models
- Volatile requirements. Requirements which change during development or when the system is in use. In a hospital, requirements derived from health-care policy

Classification of requirements

- Mutable requirements
  - Requirements that change due to the system's environment
- Emergent requirements
  - Requirements that emerge as understanding of the system develops
- Consequential requirements
  - Requirements that result from the introduction of the computer system
- Compatibility requirements
  - Requirements that depend on other systems or organizational processes

Traceability

- Traceability is concerned with the relationships between requirements, their sources and the system design
- Source traceability
  - Links from requirements to stakeholders who proposed these requirements
- Requirements traceability
  - Links between dependent requirements
- Design traceability
  - Links from the requirements to the design

A traceability matrix

Req id	1.1	1.2	1.3	2.1	2.2	2.3	3.1	3.2
1.1			R					
1.2		U	U			R		U
1.3	R			R				
2.1			R		U			U
2.2								U
2.3		R		U				
3.1								R
3.2							R	

Requirements change management

- Should apply to all proposed changes to the requirements
- Principal stages
  - Problem analysis. Discuss requirements problem and propose change
  - Change analysis and costing. Assess effects of change on other requirements
  - Change implementation. Modify requirements document and other documents to reflect change

Requirements change management