

Requirements Management

Prof. Dr. Armin B. Cremers

Overview

- What about Requirements Management
- Processes concerned with Requirements Management
 - ◆ Changes
 - ◆ Traceability of Requirements and Changes
- Tool Support
- ... few more real life scenarios ;-)

What is Requirements Management

- The process of managing the changes to requirements
- Hard Problem because of continuous changes during development process
- The principal concerns are
 - ◊ Managing the relationships between requirements
 - ◊ Managing priorities between requirements
 - ◊ Managing the dependencies between different documents
 - ➔ requirements document
 - ➔ specification
 - ➔ and other documents produced in the systems engineering process
 - ◊ Managing changes to agreed requirements
- Requirements traceability is important

Traceability

- Requirements cannot be managed effectively without traceability
- traceable means knowing about
 - ◊ who suggested the requirement
 - ◊ why exists the requirement
 - ◊ what requirements are related to it, and
 - ◊ how relates that requirement to other information such as
 - ➔ system design
 - ➔ implementations,
 - ➔ and user documentation

CASE tools for requirements management

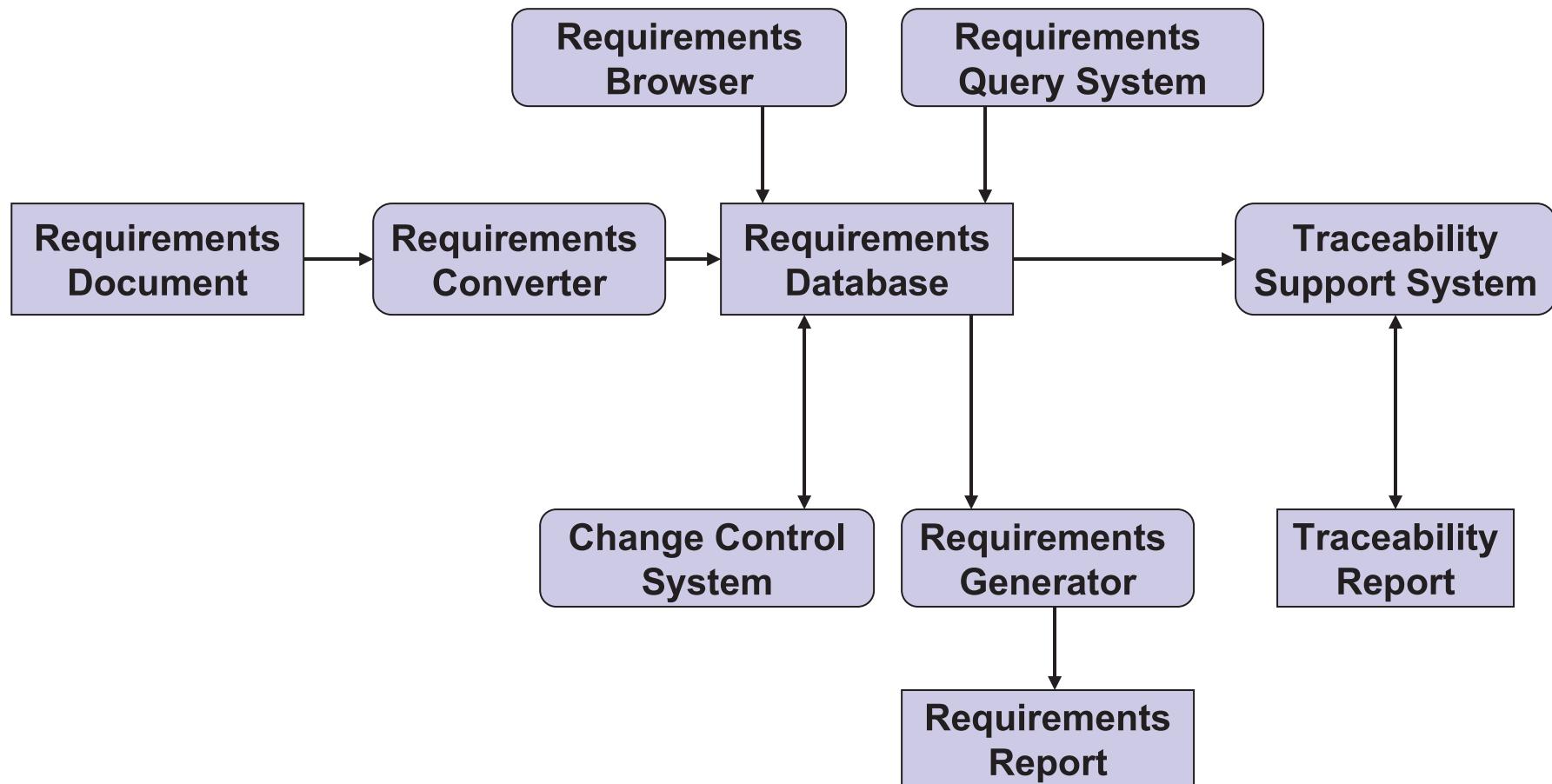
- Requirements management involves
 - ◆ collection,
 - ◆ storage,
 - ◆ maintenance of large amounts of information
- Different CASE tools available which are specifically designed to support requirements management
- Other CASE tools such may be adapted for requirements engineering
 - ◆ Configuration management systems (e.g. CVS)
 - ◆ Email (e.g. Outlook)
 - ◆ Shared workspaces (e.g. BSCW) Basic Support for Cooperative Work

Requirements management tool support - Functions

- A database system for storing requirements.
- Document analysis and generation facilities to help
 - ◊ construct a requirements database
 - ◊ create requirements documents
- Change management facilities which help ensure that changes are properly assessed
- Traceability facilities which help requirements engineers find dependencies between system requirements

A requirements management system

An ideal model



Stable and volatile requirements

- Requirements changes occur
 - ◆ while the requirements are being elicited, analyzed and validated
 - ◆ after the system has gone into service
- Some requirements are usually more subject to change than others
 - ◆ Stable requirements: Are concerned with the essence of a system and its application domain
 - ◆ Volatile requirements: Are specific to the instantiation of the system in a particular environment and for a particular customer

Requirements change factors 1/2

- Requirements errors, conflicts and inconsistencies
 - ◆ during analysis (validation, development)
 - ◆ errors and inconsistencies must be corrected
- Evolving customer/end-user knowledge of the system
 - ◆ customers and end-users develop a better understanding of what they really require from a system
- Technical, schedule or cost problems
- Changing customer priorities
 - ◆ changing business environment
 - ◆ the emergence of new competitors, staff changes, etc.
 - ◆ new laws, regulations

Requirements change factors 2/2

- Environmental changes
 - ◆ The environment in which the system is to be installed may change so that the system requirements have to change to maintain compatibility
- Organizational changes
 - ◆ changes in structure and processes
- New technology (technology push)

Types of volatile requirements

- Mutable requirements
 - ◆ due to changes to the environment in which the system is operating
- Emergent requirements
 - ◆ cannot be completely defined when the system is specified
 - ◆ emerge as the system is designed and implemented
 - ◆ emerge as users have contact with new system
- Consequential requirements
 - ◆ based on wrong assumptions about how the system will be used
 - ◆ some may be wrong
- Compatibility requirements
 - ◆ dependent on other equipment or processes
- → Best Practice: anticipate likely requirements changes (Design?)

Requirements identification

- requirements need unique identification
 - ◊ essential for requirements management
- Basic approach: requirements are numbered based on chapter/section in the requirements document
- Problems with this are:
 - ◊ Numbers cannot be unambiguously assigned until the document is complete
 - ◊ Assigning chapter/section numbers is an implicit classification of the requirement.
 - ➔ Relationships between requirements due to their “neighborhood”?
 - ➔ Misleading
 - ◊ References are hard to handle

Requirements identification techniques

- Dynamic renumbering
 - ◊ Some word processing systems allow for automatic renumbering (paragraphs, cross-references)
 - ◊ Automatic renumbering requirements depending on chapter, section and position within the section
- Symbolic identification
 - ◊ Requirements can be identified by giving them a symbolic name
 - ◊ I.e. EFF-1, EFF-2, EFF-3 are requirements which relate to system efficiency
 - ◊ Problems:
 - ➔ Meaningful Mnemonics
 - ➔ Organization of mnemonics
- Database record identification
 - ◊ Requirements are held as data in a database
 - ◊ Unique identifier per item
 - ◊ Problems:
 - ➔ External text generation
 - ➔ Highly structured

Storing requirements

- Requirements have to be stored in such a way that they can be easily
 - ◆ accessed
 - ◆ changed
 - ◆ linked (with other requirements)
 - ◆ described (in text as well as in graphics...)
 - ◆ enhanced (by adding external information)
- Possible storage techniques are
 - ◆ In one or more word processor files
 - ➔ requirements are stored in the requirements document
 - ◆ In a specially designed requirements database

Storing requirements: Word processor documents

- Advantages
 - ◆ Easy to construct, maintain, cheap
 - ◆ Requirements may be accessed by anyone with the right word processor
 - ◆ Requirements can be described informal, unstructured...
 - ◆ It is easy to produce the final requirements document
- Disadvantages
 - ◆ Requirements dependencies must be externally maintained
 - ◆ Search facilities are limited
 - ◆ Not possible to link requirements with proposed requirements changes
 - ◆ Not possible to have version control on individual requirements (only whole document)
 - ◆ No automated navigation from one requirement to another
(Improvement: Hypertext documents)

Storing requirements: Requirements database

- Each requirement is represented as one or more database entities
- Database query language is used to access requirements
- Advantages
 - ◆ Good query and navigation facilities
 - ◆ Support for change and version management
 - ◆ Versioning of single requirements
- Disadvantages
 - ◆ Readers may not have the software/skills to access the requirements database
 - ◆ Higher costs

Object classes for requirements DB

SYS_MODELS
Model: MODEL Description: TEXT Next: MODEL NULL

REQUIREMENT
Identifier: TEXT Statement: TEXT GRAPHIC Date_entered: DATE Date_changed: DATE Sources: SOURCE_LIST Rationale: REQ_RATIONALE Status: STATUS Dependents: REQ_LIST Is_dependent_on: REQ_LIST Model_links: SYS_MODELS Comments: TEXT

SOURCE_LIST
People: TEXT Documents: TEXT Reqs: REQ_LIST

REQ_LIST
Req: REQUIREMENT Description: TEXT Next: REQUIREMENT NULL

REQ_RATIONALE
Rationale: TEXT Diagrams: GRAPHIC Photos: PICTURE

Requirements DB: Choice factors 1/2

- The statement of requirements
 - ◆ text, graphics, photos
 - ◆ → external linked storage or multimedia database
- The number of requirements
- Teamwork, team distribution and computer support
 - ◆ Distributed team of requirements engineers
 - ◆ → Remote, multi-site access
 - ◆ → Browser interface

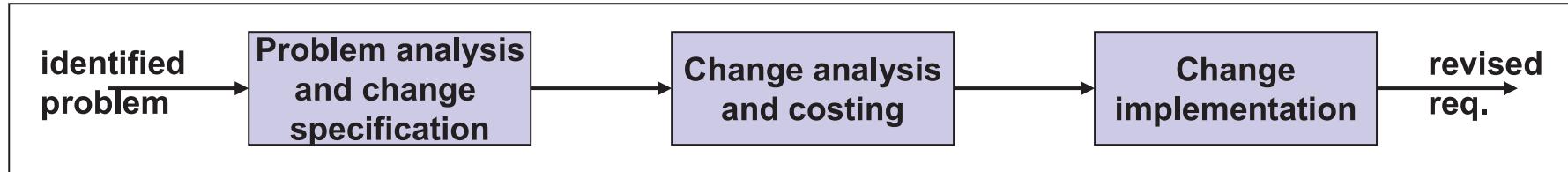
Requirements DB: Choice factors 2/2

- CASE tool use
 - ◆ The database should be the same as or compatible with CASE tool databases
 - ◆ Interfaces to CASE tools needed in later process steps
- Existing database usage
 - ◆ If a database for software engineering support is already in use, this should be used for requirements management
 - ➔ Costs of training, supporting staff, etc.

Change management

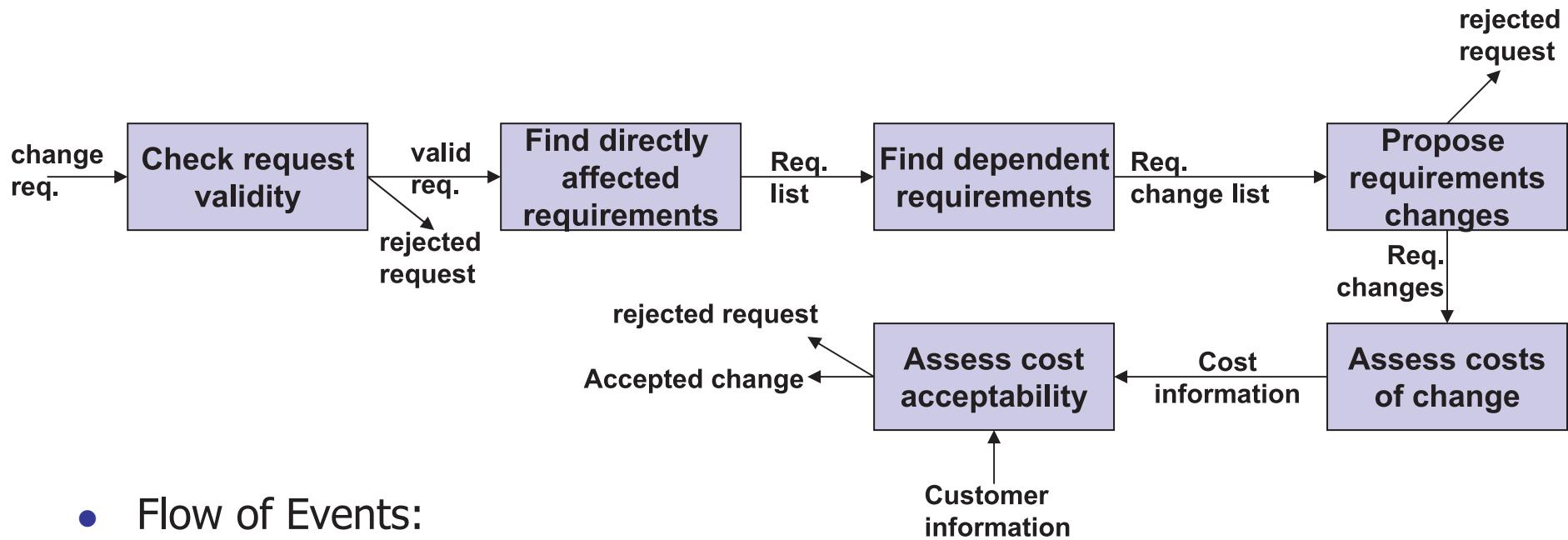
- Change management is concerned with procedures, processes and standards which are used to manage changes to system requirements
- Change *management policies* may cover:
 - ◊ Change request process
 - ◊ The information required to process each change request
 - ◊ The process used to analyze the *impact and costs* of change and the associated traceability information
 - ◊ Change Request Board (should be independent)
 - ◊ The software support (if any) for the change control process

Change management: The change management process



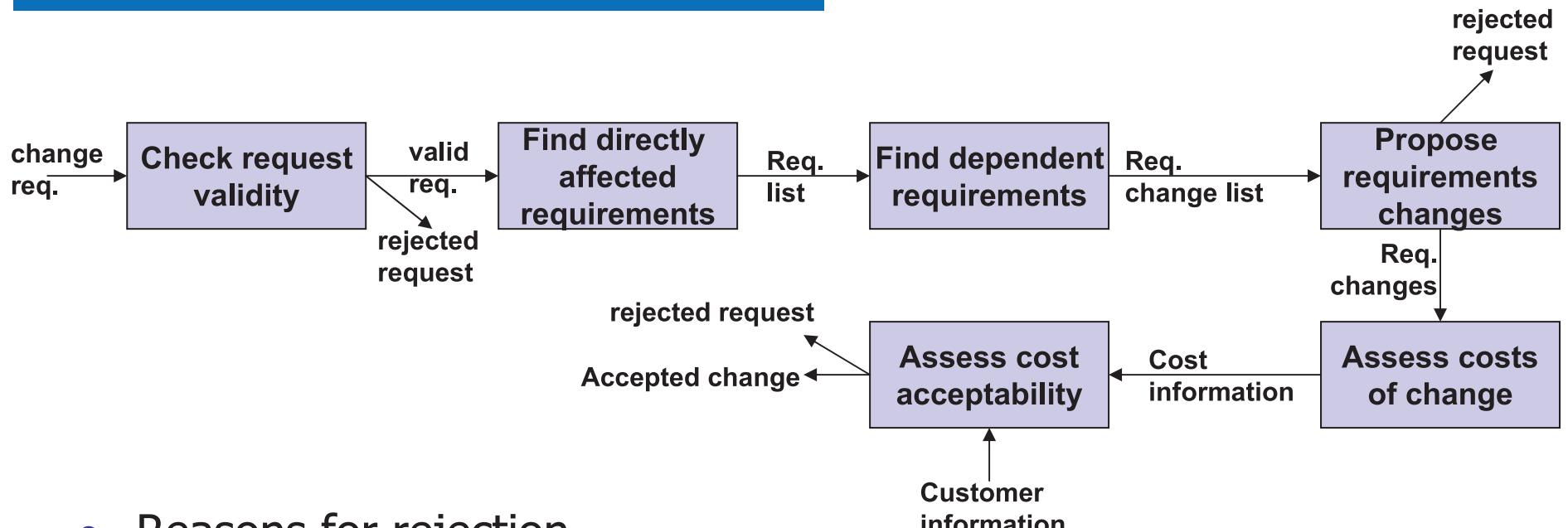
1. Identifying requirements problem
 - caused by analysis of the requirements, new customer needs, or operational problems
 - requirements changes are proposed (specified)
2. Analyzing proposed changes
 - check how many requirements (and, if necessary, system components) are affected
 - time and money, to make the change.
3. Implementing changes
 - A set of modifications to the requirements document
 - or a new document version
 - has to be validated (quality checking procedures)

Change management: Change analysis and costing 1/2



- Flow of Events:
 - ◆ Check for validity
 - Customers can misunderstand requirements and suggest unnecessary changes
 - ◆ Requirements directly affected by the change are discovered
 - ◆ Dependent requirements are discovered (using Traceability Information)
 - ◆ Propose actual changes (consultation with customer)
 - ◆ Estimating costs of making the changes
 - ◆ Negotiations with customers
 - Are the costs acceptable?

Change management: Change analysis and costing 2/2



- Reasons for rejection
 - ◆ Change request is invalid: customer has misunderstood some requirements, proposed change isn't necessary
 - ◆ Too many dependent requirements: consequential changes are unacceptable to the user
 - ◆ Costs are too high or take too long

Change processing

- Proposed changes are usually recorded on a change request form which is then passed to all of the people involved in the analysis of the change
- Change request forms may include
 - ◆ fields to document the change analysis
 - each stage of analysis
 - ◆ date fields
 - ◆ responsibility fields
 - ◆ status field
 - rejected, under consideration, accepted
 - ◆ comments field

Tool support for change management

- May be provided through requirements management tools or through configuration management tools
- Tool facilities may include
 - ◊ Electronic change request forms
 - ◊ A database to store and manage these forms
 - ◊ Group support:
 - ➔ Electronic transfer of forms between people with different responsibilities
 - ➔ electronic mail notification
 - ◊ Document Management Support:
 - ➔ direct links to a requirements database
 - ➔ Affected requirements
 - ➔ Automatic Updates

Traceability

- Traceability information
 - ◆ helps you assess the impact of requirements change
 - ◆ links related requirements and the requirements and other system representations
- Types of traceability information (Davis, 1993)
 - ◆ Backward-from traceability: Links requirements to their sources in other documents or people
 - ◆ Forward-from traceability: Links requirements to the design and implementation components
 - ◆ Backward-to traceability: Links design and implementation components back to requirements
 - ◆ Forward-to traceability: Links other documents (which may have preceded the requirements document) to relevant requirements.



Types of traceability 1/2

- Requirements-sources traceability
 - ◆ Links the requirement and the people or documents which specified the requirement
- Requirements-rationale traceability
 - ◆ Links the requirement with a description of why that requirement has been specified.
- Requirements-requirements traceability
 - ◆ Links requirements with other requirements which are, in some way, dependent on them. This should be a two-way link (dependants and is-dependent on).

Types of traceability 2/2

- Requirements-architecture traceability
 - ◆ Links requirements with the sub-systems where these requirements are implemented. This is particularly important where sub-systems are being developed by different sub-contractors
- Requirements-design traceability
 - ◆ Links requirements with specific hardware or software components in the system which are used to implement the requirement
- Requirements-interface traceability
 - ◆ Links requirements with the interfaces of external systems which are used in the provision of the requirements

Traceability tables

- Show the relationships between requirements or between requirements and design components
- Requirements are listed along the horizontal and vertical axes and relationships between requirements are marked in the table cells
- Traceability tables for showing requirements dependencies should be defined with requirement numbers used to label the rows and columns of the table

Depends-on

	R1	R2	R3	R4	R5	R6
R1			*	*		
R2					*	*
R3				*	*	
R4		*				
R5						*
R6						

Traceability lists

- Used for a small number of requirements (<250)
 - ◊ Simple tables
 - ◊ traceability tables can be implemented using a spreadsheet
- Problems:
 - ◊ Traceability becomes hard if too many requirements are available and table becomes too populated

Requirement	Depends-on
R1	R3, R4
R2	R5, R6
R3	R4, R5
R4	R2
R5	R6

Traceability policies

- Traceability policies define what and how traceability information should be maintained
- Traceability policies may include
 - ◆ The traceability information which should be maintained
 - ◆ Techniques, such as traceability matrices, which should be used for maintaining traceability
 - ◆ A description of when the traceability information should be collected during the requirements engineering and system development processes.
 - ◆ The roles of the people, such as the traceability manager, who are responsible for maintaining the traceability information should also be defined.
 - ◆ The process of managing traceability information

Factors influencing traceability policies

1/2

- Number of requirements
 - ◆ The greater the number of requirements, the more the need for formal traceability policies
- Estimated system lifetime
 - ◆ More comprehensive traceability policies should be defined for systems which have a long lifetime

Factors influencing traceability policies

2/2

- Project team size and composition
 - ◆ With a small team, it may be possible to assess the impact of proposed informally without structured traceability information. With larger teams, however, you need more formal traceability policies.
- Type of system
 - ◆ Critical systems such as hard real-time control systems or safety-critical systems need more comprehensive traceability policies than non-critical systems.
- Specific customer requirements
 - ◆ Some customers may specify that specific traceability information should be delivered as part of the system documentation.

Summary 1/2

- Requirements change is inevitable
 - ◊ customers develop a better understanding of their real needs
 - ◊ political, organizational and technical environment changes
- Stable Requirements
 - ◊ Describing the essence of a system
- Volatile Requirements
 - ◊ more concerned with how the system is implemented in a particular environment.
 - ◊ Types of volatile requirements:
 - ➔ mutable requirements
 - ➔ emergent requirements
 - ➔ consequential requirements
 - ➔ compatibility requirements.
- Unique identification of requirements is needed
- Storage of requirements: Text document, database, Hypertext documents

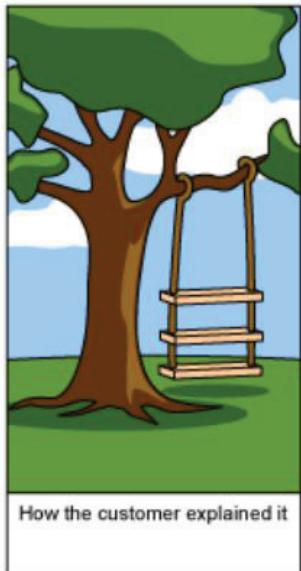
Summary 2/2

- Change management
 - ◆ defined process
 - ◆ information associated with each change request
 - ◆ responsibilities
 - ◆ Tool support
- Traceability information: Records the dependencies between
 - ◆ requirements and the sources of these requirements
 - ◆ requirements
 - ◆ requirements and the system implementation.
- Traceability in Practice
 - ◆ Traceability matrices
 - ◆ traceability policies

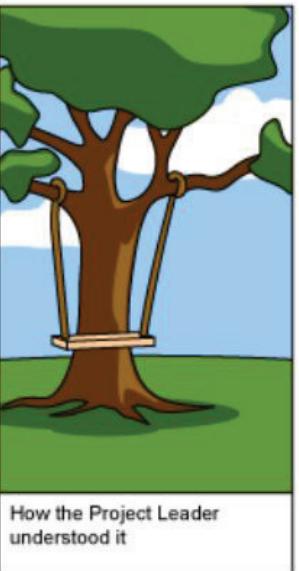
A real life scenario including many stakeholders



A real life scenario for good requirements engineering



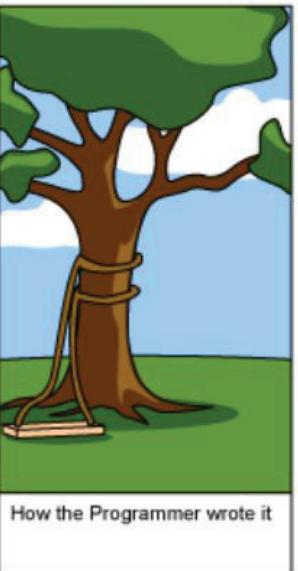
How the customer explained it



How the Project Leader understood it



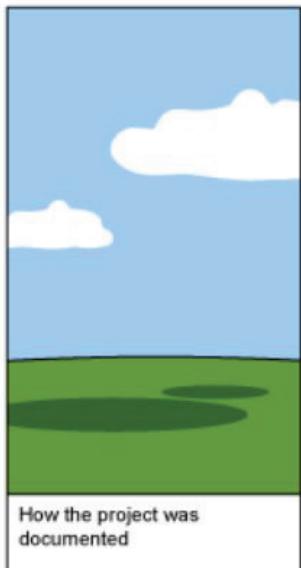
How the Analyst designed it



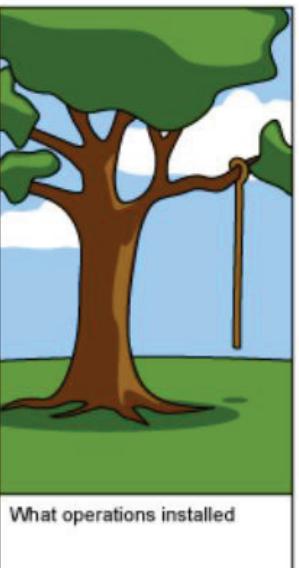
How the Programmer wrote it



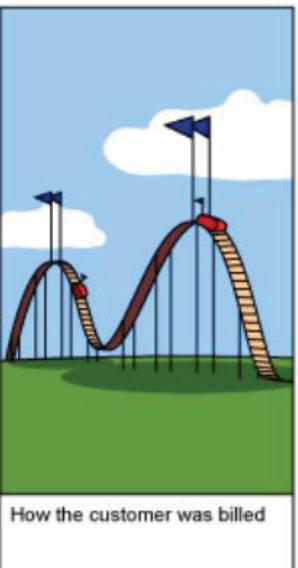
How the Business Consultant described it



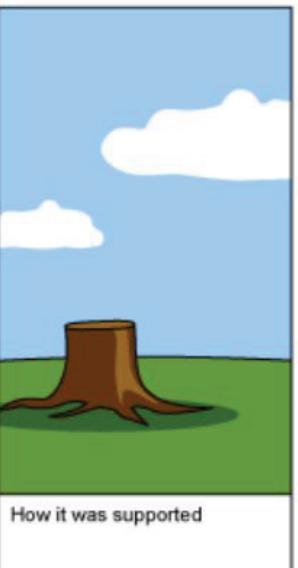
How the project was documented



What operations installed



How the customer was billed



How it was supported



What the customer really needed

