Requirements Engineering

Chapter 5 Requirements Management

Learning Objective

...to describe the process of managing the evolution and change of a system's requirements. Learning about the user (customers) needs and helping them to better understand and clarify the real and important facets of the endeavor. Some of the issues and aspects that are covered her include: *Stable* versus *volatile requirements*, *requirements identification* and *storage*, *change management* and last (but not least) *traceability*.

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

Requirements management

- The process of managing change to the requirements for a system
- The principal concerns of requirements management are:
  - Managing changes to agreed requirements
  - Managing the relationships between requirements
  - Managing the dependencies between the requirements document and other documents produced in the systems engineering process
- Requirements cannot be managed effectively without requirements traceability.
  - A requirement is traceable if you can discover who suggested the requirement, why the requirement exists, what requirements are related to it and how that requirement relates to other information such as systems designs, implementations and user documentation.

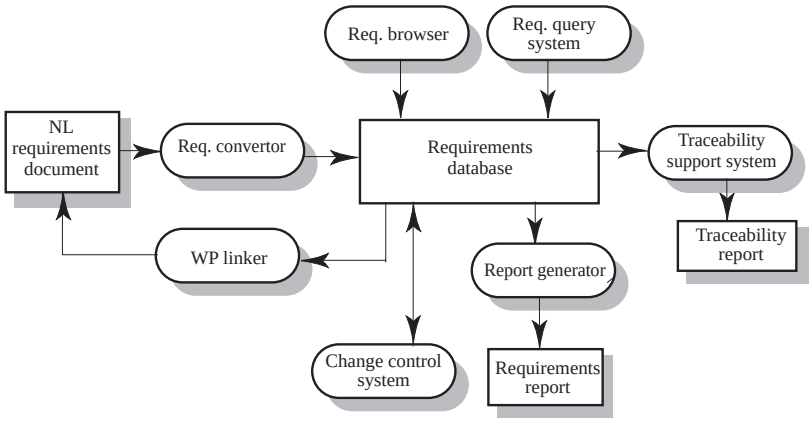
CASE tools for requirements management

- Requirements management involves the collection, storage and maintenance of large amounts of information
- There are now a number of CASE tools available which are specifically designed to support requirements management
- Other CASE tools such as configuration management systems may be adapted for requirements engineering

Requirements management tool support

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

A requirements management system



Stable and volatile requirements

- Requirements changes occur while the requirements are being elicited, analyzed and validated and 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. They change more slowly than volatile requirements.
  - Volatile requirements are specific to the instantiation of the system in a particular environment and for a particular customer.

Requirements change factors

- Requirements errors, conflicts and inconsistencies
  - As requirements are analyzed and implemented, errors and inconsistencies emerge and must be corrected. These may be discovered during requirements analysis and validation or later in the development process.
- Evolving customer/end-user knowledge of the system
  - As requirements are developed, customers and end-users develop a better understanding of what they really require from a system.
- Technical, schedule or cost problems
  - Problems may be encountered in implementing a requirement. It may be too expensive or take too long to implement certain requirements.

Requirements change factors

- Changing customer priorities
  - Customer priorities change during system development as a result of a changing business environment, the emergence of new competitors, staff changes, etc.
- 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
  - The organization which intends to use the system may change its structure and processes resulting in new system requirements

Types of volatile requirement

- Mutable requirements
  - These are requirements which change because of changes to the environment in which the system is operating
- Emergent requirements
  - These are requirements which cannot be completely defined when the system is specified but which emerge as the system is designed and implemented
- Consequential requirements
  - These are requirements which are based on assumptions about how the system will be used. When the system is put into use, some of these assumptions will be wrong.
- Compatibility requirements
  - These are requirements which depend on other equipment or processes.

Requirements identification

- It is essential for requirements management that every requirement should have a unique identification
- The most common approach is requirements numbering 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. This can mislead readers of the document into thinking that the most important relationships are with requirements in the same section

Requirements identification techniques

- Dynamic renumbering
  - Some word processing systems allow for automatic renumbering of paragraphs and the inclusion of cross-references. As you re-organize your document and add new requirements, the system keeps track of the cross-reference and automatically renumbers your requirement depending on its chapter, section and position within the section
- Database record identification
  - When a requirement is identified it is entered in a requirements database and a database record identifier is assigned. This database identifier is used in all subsequent references to the requirement
- Symbolic identification
  - Requirements can be identified by giving them a symbolic name which is associated with the requirement itself. For example, EFF-1, EFF-2, EFF-3 may be used for requirements which relate to system efficiency

Storing requirements

- Requirements have to be stored in such a way that they can be accessed easily and related to other system requirements
- Possible storage techniques are
  - In one or more word processor files - requirements are stored in the requirements document
  - In a specially designed requirements database

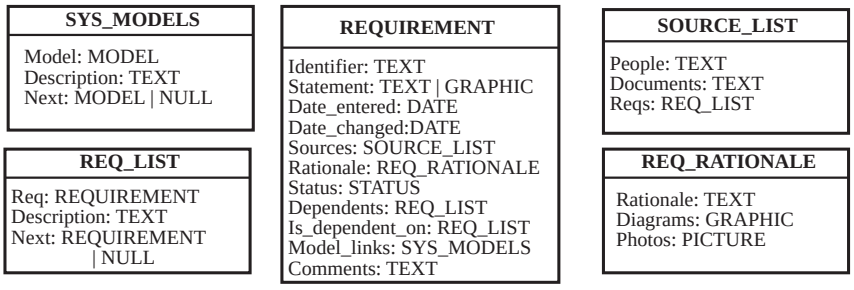
Word processor documents

- ⊗ Advantages
  - Requirements are all stored in the same place
  - Requirements may be accessed by anyone with the right word processor
  - 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
  - No automated navigation from one requirement to another

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
- ⊗ Disadvantages
  - Readers may not have the software/skills to access the requirements database
  - The link between the database and the requirements document must be maintained

Object classes for requirements DB



Requirements DB - choice factors

- ⊗ The statement of requirements
  - If there is a need to store more than just simple text, a database with multimedia capabilities may have to be used.
- ⊗ The number of requirements
  - Larger systems usually need a database which is designed to manage a very large volume of data running on a specialized database server.
- ⊗ Teamwork, team distribution and computer support
  - If the requirements are developed by a distributed team of people, perhaps from different organizations, you need a database which provides for remote, multi-site access.

Database choice factors

- CASE tool use
  - The database should be the same as or compatible with CASE tool databases. However, this can be a problem with some CASE tools which use their own proprietary database
- Existing database usage
  - If a database for software engineering support is already in use, this should be used for requirements management.

Change management

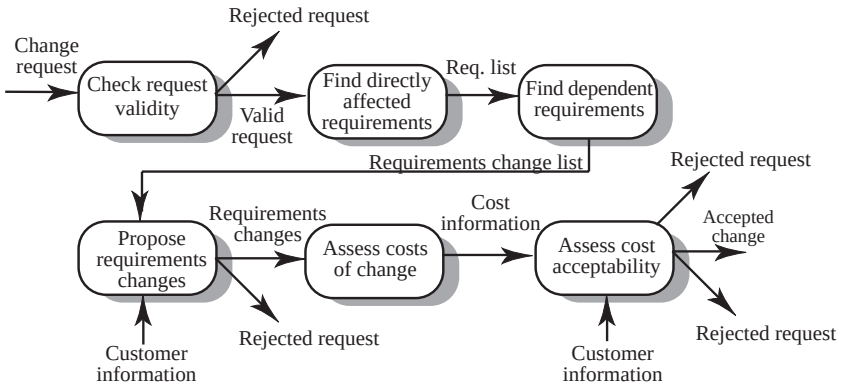
- Change management is concerned with the procedures, processes and standards which are used to manage changes to system requirements
- Change management policies may cover:
  - The change request process and the information required to process each change request
  - The process used to analyze the impact and costs of change and the associated traceability information
  - The membership of the body which formally considers change requests
  - 4. The software support (if any) for the change control process

The change management process

- Some requirements problem is identified.
  - This could come from an analysis of the requirements, new customer needs, or operational problems with the system. The requirements are analyzed using problem information and requirements changes are proposed.
- The proposed changes are analyzed
  - This checks how many requirements (and, if necessary, system components) are affected by the change and roughly how much it would cost, in both time and money, to make the change.
- The change is implemented.
  - A set of amendments to the requirements document or a new document version is produced. This should, of course, be validated using whatever normal quality checking procedures are used.

Change management stages

Change analysis and costing



Change analysis activities

- The change request is checked for validity. Customers can misunderstand requirements and suggest unnecessary changes.
- The requirements which are directly affected by the change are discovered.
- Traceability information is used to find dependent requirements affected by the change.
- The actual changes which must be made to the requirements are proposed.
- The costs of making the changes are estimated.
- Negotiations with customers are held to check if the costs of the proposed changes are acceptable.

Change request rejection

- If the change request is invalid. This normally arises if a customer has misunderstood something about the requirements and proposed a change which isn't necessary.
- If the change request results in consequential changes which are unacceptable to the user.
- If the cost of implementing the change is too high or takes 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
  - data fields
  - responsibility fields
  - status field
  - comments field

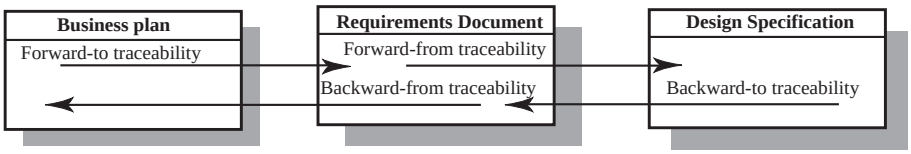
Tool support for change management

- ⊗ May be provided through requirements management tools or through configuration management tools
- ⊗ Tool facilities may include
  - Electronic change request forms which are filled in by different participants in the process.
  - A database to store and manage these forms.
  - A change model which may be instantiated so that people responsible for one stage of the process know who is responsible for the next process activity.
  - Electronic transfer of forms between people with different responsibilities and electronic mail notification when activities have been completed.
  - In some cases, direct links to a requirements database.

Traceability

- ⊗ Traceability information is information which helps you assess the impact of requirements change. It links related requirements and the requirements and other system representations
- ⊗ Types of traceability information
  - *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 backs to requirements
  - *Forward-to traceability* Links other documents (which may have preceded the requirements document) to relevant requirements.

Backwards/forwards traceability



Types of traceability

- ⊗ 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

- 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

- 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

A traceability table

Depends-on

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

Traceability lists

- If a relatively small number of requirements have to be managed (up to 250, say), traceability tables can be implemented using a spreadsheet
- Traceability tables become more of a problem when there are hundreds or thousands of requirements as the tables become large and sparsely populated
- A simplified form of traceability table may be used where, along with each requirement description, one or more lists of the identifiers of related requirements are maintained.
- Traceability lists are simple lists of relationships which can be implemented as text or as simple tables

A traceability list

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.
  - A description of how to handle and document policy exceptions
  - The process of managing traceability information

Factors influencing traceability policies

- 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.
- Level of organizational maturity
  - Detailed traceability policies are most likely to be cost-effective in organizations which have a higher level of process maturity

Factors influencing traceability policies

- 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.

Key points

- Requirements change is inevitable as customers develop a better understanding of their real needs and as the political, organizational and technical environment in which a system is to be installed changes.
- Requirements which are concerned with the essence of a system are more likely to be stable than requirements which are more concerned with how the system is implemented in a particular environment.
- Types of volatile requirement include mutable requirements, emergent requirements, consequential requirements and compatibility requirements.
- Requirements management requires that each requirement should be uniquely identified.
- If a large number of requirements have to be managed, the requirements should be stored in a database and links between related requirements should be maintained.

Key points

- Change management policies should define the processes used for change management and the information which should be associated with each change request. They should also define who is responsible for doing what in the change management process.
- Some automated support for change management should be provided. This may come through specialized requirements management tools or by configuring existing tools to support change management
- Traceability information records the dependencies between requirements and the sources of these requirements, dependencies between requirements and dependencies between the requirements and the system implementation.
- Traceability matrices may be used to record traceability information.
- Collecting and maintaining traceability information is expensive. To help control these costs, organizations should define a set of traceability policies which set out what information is to be collected and how it is to be maintained.