# Requirements Management
## (3C05/D22)

ComputerScience

---

## Unit 10: Requirements Management

- Objectives
  - To introduce requirements management and its role within the development process.
  - To examine the support available for requirements management.

ComputerScience

---

## A Brief Reminder

- requirement - in system/software engineering:
  - a capability needed by a user to solve a problem or achieve an objective;
  - a capability that must be met or possessed by a system or system component to satisfy a contract, standard, specification or other formally imposed document;
  - the set of all requirements that form the basis for subsequent development of the software or software component;
  - short description sometimes used in place of the term software requirements specification.

ComputerScience

## Requirements Elicitation & Specification

- This unit assumes that you are able to elicit (alternative terms are capturing or gathering) requirements. This is by no means easy or straightforward. There are a large body of techniques associated with doing this.

  such as structured interviews, questionnaires, observation, knowledge acquisition

- It also assumes that you are able to represent and document these requirements in a requirements specification (or specifications).

  generally natural language, carefully structured and attributed, complemented by appropriate models

**Computer**Science

## A Common Mistake

- A common mistake is to think of requirements engineering (the term that embraces requirements elicitation, specification and management) as concerning the front-end of the lifecycle.
- Requirements engineering carries on for the whole life of the system. It focuses on ensuring that "the voice of the customer" is heard at all points in the development process from the initial conception of the system, through design, testing and changes introduced for maintenance and system evolution.

  VERY IMPORTANT!

**Computer**Science

## Requirements Management

- Requirements management is a new term which has been rapidly adopted by industry. It is the activity concerned with the effective control of information related to system requirements and in particular the preservation of the integrity of that information for the life of the system and with respect to changes in the system and its environment.

**Computer**Science

## Why is Requirements Management Important?

- systems continue to be built which do not meet user needs hence quality-oriented approaches to development which involve specifying user and quality requirements and using these requirements to drive, control and evaluate the development process
- this depends on
- the ability to establish and maintain a connection between the information that has been elicited as needs, the requirements derived from these and the subsequent artefacts in which these requirements are realised

and critically, to continue to do so in the face of inevitable requirements change

**Computer**Science

## The Bottom Line!

- quality makes no sense without reference to requirements

- quality-oriented development is requirements-driven development

- requirements management is a prerequisite for quality-oriented development

In any case requirements management is *required* by ISO9000, CMM and most large system procurers
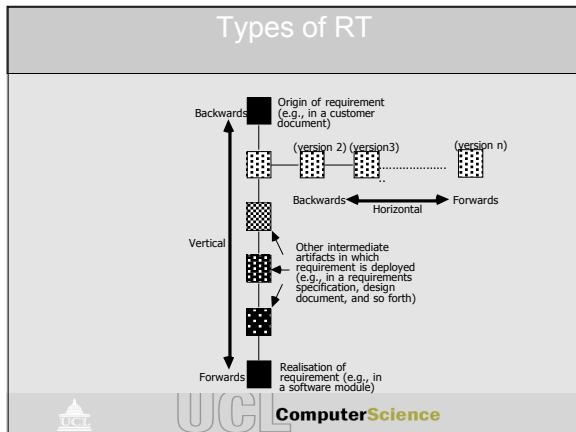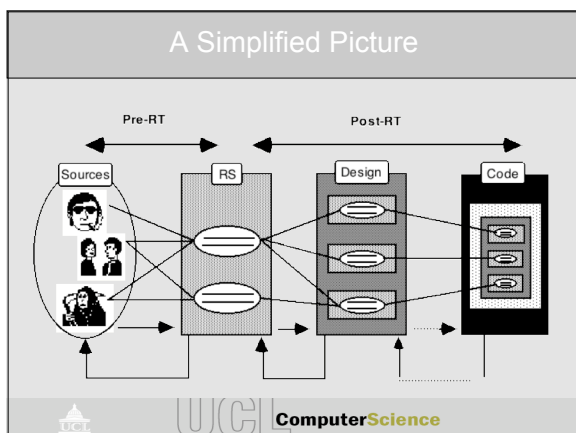
**Computer**Science

## Requirements Traceability

- Requirements traceability is at the heart of requirements management. Requirements traceability (abbreviated, RT) refers to the ability to describe and follow the life of a requirement in both a forwards and backwards direction (ie from its origins, through its development and specification, to its subsequent deployment and use, and through periods of ongoing refinement and iteration in any of these phases).

**Computer**Science

## Types of RT



## Types of RT

- Pre-requirements traceability (pre-RT) refers to the ability to describe and follow those aspects of a requirement's life prior to its inclusion in the RS in both a forwards and backwards direction (i.e., requirements production and refinement).
- Post-requirements traceability (post-RT) refers to the ability to describe and follow those aspects of a requirement's life that result from its inclusion in the RS in both a forwards and backwards direction (i.e., requirements deployment and use).

## A Simplified Picture

## Support for RT

- techniques are explicit mechanisms through which RT can be achieved
- approaches are organised systems and software development practices which incorporate techniques to support requirements traceability or in which requirements traceability is a by product of their use
- automated tools embed support for requirements traceability

**Computer**Science

## Techniques

- cross-reference centred
  - simple
  - hypertext supported
  - tagging, numbering & indexing
  - traceability matrices and matrix sequences

  *widely used in industry, basic good practice*

- document-centred
  - document templates
  - integration/transformation documents

  *emerging from research*

- structure-centred
  - truth maintenance networks
  - constraint networks and propagation

  *"left field" but interesting*

**Computer**Science

## Approaches

- models

  *basic good practice*

  - traceability support through development process models

- methods

  *basic good practice*

  - traceability support through orderly development of related artefacts

- languages

  *experimental*

  - traceability support through languages with built in traceability constructs

**Computer**Science

## Automated Tools

| | strengths | weaknesses |
|---|---|---|
| • general purpose tools<br>  – e.g. wp, spreadsheets, hypertext editors, databases | flexible<br>easily available<br>small projects | high start-up cost<br>difficult to maintain<br>unpredictable |
| • workbenches<br>  – CASE work benches<br>    – e.g. Rational Rose | tight<br>by-product | rigid<br>limited |
|   – dedicated RT workbenches<br>    – e.g. DOORS, RTM, RDD-100 | fine-grained<br>added value | RT becomes focus<br>depends on buy-in<br>varies bn phases |
| • - environments<br>  – e.g. Rational Suite | full lifecycle<br>all artefacts<br>can be distributed | coarse grain<br>backwards weak |

## DOORS

**example**

- DOORS is popular for a number of reasons, including:
- (1) The ability to make arbitrary traceability links between information sets. The ensuing ability to construct hierarchies of heterogeneous types of document means it handles the RT in large projects by decomposing their documents into lots of smaller ones and managing the interactions between them.
- (2) The ability to integrate with a number of third-party tools to support other development activities. In addition, through the provision of an open tool interface builder, it can further be configured to integrate with customer's own in-house tools.

## DOORS

**example**

- (3) The provision of a scripting language, the DOORS Extension Language (DXL), which enables the functionality of the tool to be extended and customised. Libraries of useful RT-related functions, such as the costing of requirements changes, can be developed.
- (4) It does not require much expertise and lengthy training to use the basic features of DOORS, though increased utility does come from learning to use DXL.

## RTM

- RTM is popular for a number of reasons, including:
- (1) The ability to be pre-configured to address different project RT needs and project lifecycles. This means that it does not impose strict pre-conditions on use, but guides use once the RT scheme to be used has been configured using its graphical schema definition facility.
- (2) The ability to interface directly with a number of third-party tools to support other development activities, most notably with RDD-100. This again offers the potential for lifecycle-wide RT.

**Computer**Science

## RTM (Continued)

- (3) The ability to automatically identify and capture requirements from source documents using a sophisticated autostripper tool.
- (4) Its database partitioning option manages requirements and their traceability across disconnected networks. As this facility enables both multiple and selective partitions, it is particularly suited to the growing culture of subcontracting in industry.

**Computer**Science

## Requirements Management: a Continuing Problem

- Claims for Requirements Management that are made by tool vendors are not realised in practice.
  - This is because there are many difficult issues that need to be considered prior to using such Requirements Management tools. The problems lie firstly in setting up a shared, consistent, and coherent Requirements Management scheme for each project. They then lie in the need for one hundred percent commitment from all the stakeholders and in the need for some overall coordination.

**Computer**Science

## Technical Problems (Not)

- With this is mind, many of the problems still being experienced are not technical problems, but human and organisational problems. Although technical solutions are still needed for projects with huge numbers of requirements, like in the U.S. DoD, most of the outstanding problems do not have purely technical solutions.

**ComputerScience**

## Resourcing

- Requirements Management has a high start-up cost and needs continued funding throughout a project.
  - Project funding is often limited at the onset of a project, restricted to those aspects of the project which are tangible and visible, and subsequently allocated in a phase-by-phase manner.
  - This means that short-cuts are often made with Requirements Management when there are problems with budget or time.
  - In many projects, Requirements Management is not even considered until it is required to start addressing the problems that inevitably arise, by which time it is generally too late.

**ComputerScience**

## The Provider/End-user Conflict

- One party's benefits are often obtained at the other party's expense. Addressing one party's concerns often makes it problematic to address the other's.
- The two main parties involved, those who would be in a position to make Requirements Management possible and those who would subsequently require requirements related information to assist their work, have conflicting problems and needs.

Its not relevant for them!

Nobody else will do it or keep it up to date

I am too busy!

**ComputerScience**

## Key Points

- Requirements management is a critical activity for system development. It ensures that the voice of the customer is heard throughout the development process. Requirements engineering is not restricted to a single phase in the lifecycle.
- The central task of requirements management is assuring traceability of the requirements both forwards and backwards and from the earliest requirements elicitation activities through to system evolution and maintenance.
- Techniques, approaches and tools can help but ultimately requirements management depends on commitment from management and the whole project team.

**UCL** **Computer**Science