Requirements Management

Anthony Finkelstein

Professor of Software Systems Engineering

Department of Computer Science

University College London

with Olly Gotel (DERA) & Richard Stevens (QSS)

[Email] A.Finkelstein@cs.ucl.ac.uk

[URL] http://www.cs.ucl.ac.uk/staff/A.Finkelstein

Objective

This tutorial provides an overview of the "state-of-the-art" in the area of requirements management in a systems engineering setting. It provides a detailed look at requirements traceability and at practical techniques for supporting it. The tutorial will outline an action plan for improving the management of requirements in an industrial organization.

Establishing Common Ground

- *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.

A Working Assumption

warn me if this is not true!

- This tutorial 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

Orientation

- "Requirements engineering is the branch of systems engineering concerned with the real-world goals for, services provided by, and constraints on a large and complex software-intensive system. It is also concerned with the relationship of these factors to precise specifications of system behaviour, and to their evolution over time and across system families."
- Establishing the needs that have given rise to the development process and organising this information in a form that will support system conception and implementation.

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.

Why are Requirements Important?

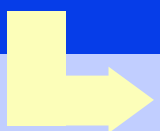
- the negative case
  - client contact
  - time and effort expended
  - error removal cost
  - risk minimisation
- the positive case
  - ensures user focus
  - supports adaptation and evolution

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

Survey - 17 countries - 4000 responses

- Software developers in the IT, production and service sectors consistently ranked "requirements specification" and "managing customer requirements" as the most important problems they faced. In the case of requirements specification more than 50% of respondents rated it as a "major problem" and 35% of respondents rated as a "minor problem". Less than 12% of respondents rated it as "never a problem", the lowest rating in the survey. Similar responses were obtained for managing customer requirements.

"Requirements were regarded as significantly more problematic than documentation, testing, quality systems, standards, design, configuration management, and programming."

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

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!

"A Large Organisation"

at our sponsors request!

- government owned (at the moment)
- provides services in the transport sector
- procures, operates and maintains very large software-intensive systems, some of which are safety-critical
- projects with a budget of up to half a billion pounds
- development processes for these systems are long-term and complex
- processes involve internal and external organisations, some of which are located in different European countries.

Other Key Features

- procurement-oriented
- Public Finance Initiative
- rebuilding business processes to develop a greater customer orientation

direct analogues of what is happening in many private sector organisations

they said

"We Want a Tool for {Requirements} Management"

we said

"Experience in software engineering has shown that simply purchasing tools rarely brings the benefits that are anticipated. Tools work when they fit with a clear, well understood and well conceived process. This is particularly important in the area of requirements management where such processes are complex, sensitive and frequently cross organisational boundaries. It applies with particular strength to requirements management tools which must ensure that the requirements are managed for the *life* of the system thus interacting with, and impacting on, every aspect of the overall development process."

Issue Classes

- Processes
  - Issues associated with this class are related to processes and standards on an organisational rather than project level.
- Requirements Content
  - Issues associated with this class are directly related to the content of requirements documents.
- Domain
  - Issues associated with this class are related to capturing the specialised domain in which the organisation operates.

Issue Classes (Continued)

- Organisational Infrastructure
  - Issues associated with this class are about organisational level support for systems engineering.
- People
  - Issues associated with this class are related to people, for example education, motivation, or training.
- Tools
  - Issues associated with this class are concerned with tools and their use.

Process-related Issues - general

- Process-driven system development.
  - Decisions on which standards, processes, technologies, tools, methodologies are used for a project are left entirely to the project manager.
- Standards, processes and support.
  - While there are some manuals primarily relating to management, standards and processes for systems engineering are not properly defined and no support for them is readily available.

Process-related Issues - general (continued)

- Ad-hoc process documentation.
  - There are no guidelines on how to document systems engineering processes.
- Process assessment.
  - System engineering processes have not been assessed nor is there a continuing programme of process assessment.

Process-related Issues - general (continued)

- Suppliers and standards.
  - Suppliers are confused by the fact that no standard processes or procedures are in place and that they may be mandated to follow different processes with different support requirements from project to project.
- Collaboration.
  - Agreement about requirements engineering processes and tools needs to be reached between collaborating parties in Europe. The position of the organisation is weakened by a lack of an established and well-documented set of process guidelines.

Contracting Process Issues

- Contractor process.
  - The degree of involvement in determining the contractor's processes is different from project to project. There is no set policy with respect to either the process or the appropriate level of involvement in determining the process.
- Contractor compliance.
  - The monitoring of compliance to processes, where these are mandated, is ad-hoc.

Contracting Process Issues (continued)

- Contractor control.
  - There are particular problems related to the control of software processes resulting from lack of immediate availability of software engineering expertise within some systems engineering projects.
- Contracting of requirements management.
  - Aspects of requirements management are increasingly being contracted. Without appropriate processes and well-developed practices in this area there is a danger that the organisation might lose control.

Stakeholder Issues

- Stakeholder identification.
  - The task of identifying stakeholders who ought to have a voice in the requirements of the system is not handled systematically.
- Stakeholder empowerment.
  - It is not sufficient to identify stakeholders. They need to be empowered to act. This includes being given resources to participate in the requirements engineering process.

Stakeholder Issues (continued)

- Stakeholder involvement.
  - The involvement of stakeholders in the requirements engineering process must be continued for the life of that process. A "one-shot" involvement cannot be effective.
- Requirements ownership.
  - No owner of a requirement is explicitly determined.

Other Process Related Issues

- Configuration management process.
  - There is no consistent configuration management process.
- Integration of prototyping.
  - A prototype can be developed leading to significant insights into usability issues but not made part of the requirements process and hence contract. For this reason the contractor didn't pay attention and the prototype was thrown away.

Other Process Related Issues (continued)

- Contract and purchasing strategy.
  - The contracts and purchasing department is generally involved in a project too late. The requirements process would benefit if a contract and purchasing strategy were identified early.
- Traceability of tender evaluation.
  - Tender evaluation is not managed alongside the requirements engineering process.

Other Process Related Issues (continued)

- Conflicting requirements.
  - There is no well-defined process for resolving conflicting requirements.
- Safety requirements.
  - Despite a good safety process, the requirements and the safety process are not tied together effectively.
- Freezing requirements.
  - It can be risky to have requirements, which are endlessly subject to change. Freezing requirements too early is equally problematic.

Requirements Content Issues

- Representation.
  - While natural language as a means of expressing requirements has many virtues, the sole reliance on natural language can lead to conflicting, incomplete or ambiguous requirements definitions.
- Requirements document templates.
  - In order to structure requirements, projects often use documents of past projects adopting the same structure. These are of variable quality. There is no guidance on structure or content of requirements documents. Nor is there any consistency across projects.

Requirements Content Issues (continued)

- Granularity - overspecification
  - The requirements for well-understood parts of the system tend to be over-specified.
- Granularity – "tip-of-iceberg" requirements
  - Single requirements statements hide huge and complex requirements without any indication, even where this problem is known.

Requirements Content Issues (continued)

- Acceptance test traceability
  - No traceability from requirements to acceptance test criteria.
- Acceptance test derivation.
  - Acceptance test criteria are not systematically derived from requirements.
- Success criteria for acceptance tests.
  - No success criteria for acceptance tests are specified. Acceptance test cases are an a-posterior deliverable of a contractor rather than an essential part of the requirements!

Requirements Content Issues (continued)

- "Solution-free" requirements.
  - When users specify requirements they very often have a certain solution of a certain supplier in mind.
- Modelling notations.
  - Stakeholders find difficulty in validating models presented directly and without interpretation in systematic notations (for example Yourdon essential models or BNF).

Requirements Content Issues (continued)

- Impact analysis.
  - There is no impact analysis. The requirements are not organised so that the impact of changing a requirement on other requirements or on the system design can be determined.
- Risk management.
  - Requirements are not classified according to the risk of not achieving them within a given budget. That risk is not managed throughout the processes.

Requirements Content Issues (continued)

- User interaction modelling.
  - Despite the fact that the class of systems dealt with by the organisation are user interaction intensive, except in limited cases, no user interaction and risk modelling is done.
- Losing rationale.
  - Rationale of requirements is not adequately documented. In particular the reasons why requirements are not included are rarely given.

Requirements Content Issues (continued)

- Relationships between requirements.
  - Relationships between requirements are not identified and maintained.
- Non-Functional-Requirements and system-wide properties.
  - The requirements specifications are organised around functional blocks. Non-functional requirements and system-wide properties are not managed or tied into the requirements process. They are frequently omitted or repeated inappropriately for each functional block.

Requirements Content Issues (continued)

- Precision vs. readability.
  - Because the natural language text is the only carrier used to express requirements information, the text is forced to be more precise than it can be naturally achieved.

Requirements Content Issues (continued)

- Prioritisation of requirements.
  - Some requirements are more important than others in terms of the benefits they deliver to stakeholders. The importance of requirements is not identified.
- Yourdon essential models.
  - The development of Yourdon essential models submerged the system architecture, which was represented by functional blocks before.

Domain-related Issues

- Concept of Operation.
  - Lack of a documented concept of operation or "domain model".
- System Architecture.
  - The system architecture for the overall service the organisation provides, regarded by all analysts as an important organising principle of requirements documents, is implicit and it is not consistently shared across projects.

Organisational Infrastructure Issues

- Expert support.
  - No requirements engineering experience is readily accessible. Information can only be obtained through informal contacts or personal relations. Skilled requirements engineers are not involved in conception stage of projects.
- Hardware and software infrastructure.
  - Systems engineering groups are equipped at a level appropriate for general office work but not provided with a hardware and software infrastructure appropriate to the work they are doing, and the tools they use.

Organisational Infrastructure Issues (Continued)

- Organisational learning and memory.
  - Knowledge that is available is not transferred onto an organisational level in order to make it accessible for upcoming projects. There is no concept of an organisational memory.
- Requirements reuse.
  - Requirements are not maintained at an organisational level that would allow reuse.

People-related Issues

- RE training.
  - No courses and training on requirements engineering and tools is available for people starting to work in this area. Training tends to be in specific methods or tools rather than primary background knowledge about the task.
- Motivation.
  - There has been insufficient effort to motivate people to use tools and techniques. There has been relatively little effort to clearly demonstrate the benefits to all those involved.

People-related Issues (Continued)

- RE awareness.
  - Senior management may not necessarily understand the requirements engineering process and the role of such matters as traceability.

Tool-related Issues

- Configuration management.
  - The use of simple word processors for requirements management in many cases makes it difficult to achieve proper configuration management.
- Requirements search mechanism.
  - Due to the lack of usage of appropriate requirements engineering support tools, no complex search can be performed on requirements documents.

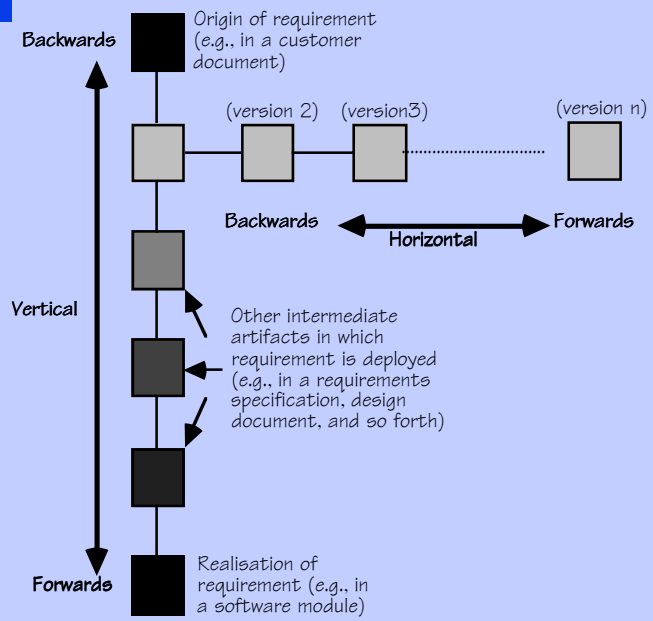
Tool-related Issues (Continued)

- Tool usability.
  - The large requirements management tool used within the organisation has significant usability drawbacks making adoption a difficult task.
- Tool usage.
  - People are not used to the style of "online computer mediated working".
- Silver bullets.
  - There is a belief in a silver bullet tool.

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).

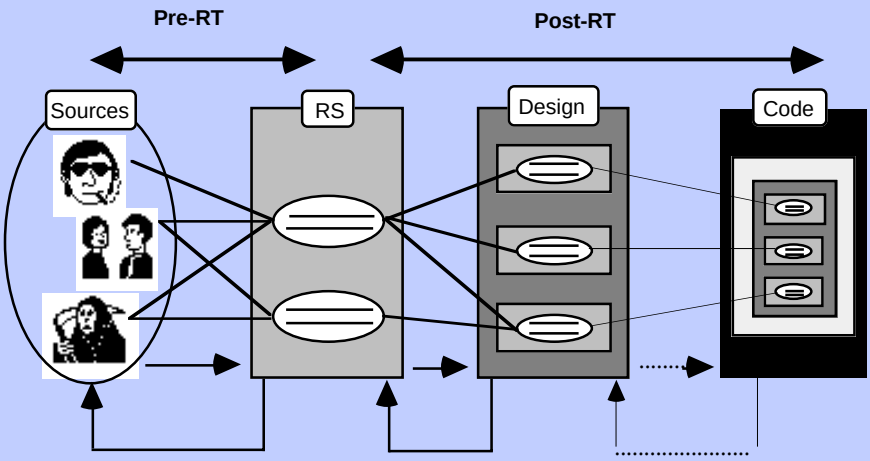
Types of RT



Types of RT (Continued)

- 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

Techniques

- cross-reference centred
  - simple
  - hypertext supported
  - tagging, numbering & indexing
  - traceability matrices and matrix sequences
- document-centred
  - document templates
  - integration/transformation documents
- structure-centred
  - truth maintenance networks
  - constraint networks and propagation

widely used in industry, basic good practice

emerging from research

"left field" but interesting

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

Automated Tools

- general purpose tools
  - e.g. wp, spreadsheets, hypertext editors, databases
- workbenches
  - CASE work benches
    - e.g. Statemate Magnum
  - dedicated RT workbenches
    - e.g. DOORS, RTM, RDD-100
- - environments
  - e.g. IEF, Cradle

strengths	weaknesses
flexible easily available small projects	high start-up cost difficult to maintain unpredictable support
tight by-product	rigid limited
fine-grained added value	RT becomes focus depends on buy-in varies bn phases
full lifecycle all artefacts can be distributed	coarse grain backwards weak

Questions to Ask

- what priority does the tool give to RT?
- what mechanical and analytical support is provided by the tool for establishing RT?
- who has to establish the RT when using the tool?
- what kind of requirements-related information can be made traceable by the tool?
- what is the breadth and longevity of the RT provided by the tool?
- what are the main tasks and job roles which are supported by the RT the job provides?

DOORS

- 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 (Continued)

(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.

we will be using DOORS in some exercises

icCONCEPT 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.

icCONCEPT 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.

Future

- short term
  - feature mimicing
  - robustness
  - move away from genericity
  - move to business applications
  - tool integration
- medium term
  - distribution
  - web integration
  - multimedia

Future

- long term
  - process integration
  - "literate" modelling
  - architecture
  - knowledge management

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.

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.

Problem Analysis

- we need to define what the RT problem actually is if we are to have any chance of identifying any potential technical and organisational solutions to address it
- we need to uncover the issues which underlie current problems, and so identify those which hold the potential for the most long-term and far-reaching improvements

RT as a "wicked problem"

(a) there is no unique solution, but any solutions that are put forward help to highlight what the real problems are;

(b) there are multiple stakeholders, though this set of stakeholders is not stable;

(c) there are no ultimate stopping conditions, so only satisficing solutions will ever be found.

Lack of a Common Definition

Purpose-Driven: definitions falling in this category define RT in terms of what it should do. Examples include:

- (1) Requirements traceability is "the means whereby software producers can 'prove' to their client that: the requirements have been understood; the product will fully comply with the requirements; and the product does not exhibit any unnecessary feature or functionality"
- (2) "Requirements traceability is the ability to adhere to the business position, project scope and key requirements that have been signed off"

Lack of a Common Definition

Solution-Driven: definitions falling in this category define RT in terms of how it should be implemented. Examples include:

- (1) "Traceability refers to the ability of tracing from one entity to another based on given semantic relations"
- (2) "Traceability refers to the ability to cross-reference items in the requirements specification with items in the design specification"

Lack of a Common Definition

Information-Driven: definitions falling in this category define RT in terms of the information that it should trace between. Examples include:

- (1) "Requirements traceability is the ability to link between functions, data, requirements and any text in the statement of requirements that refers to them"
- (2) The paragraph for requirements traceability must contain "a mapping of the engineering requirements in this (Software Requirements) Specification to the requirements applicable to this Computer Software Configuration Item in the System/Segment Specification, Prime Item Development Specification, or Configuration Item Development Specification"

Implications

- No single definition accounts for all perspectives on RT
- How can RT be consistently provided if each individual in a team, and each team in a project, have their own understanding as to what is meant by RT?
- How can techniques, approaches, or tools be coherently used together if they embed, and so support, incompatible notions of RT?
- RT problems will continue to exist in practice so long as these dispersed viewpoints are not recognised and reconciled

Diverse Sources: empirical study

(1) multiple incompatible and fragmented documents, from distributed sources, with no clear relationship to a unified requirements specification;

(2) inability to handle the increasing amounts of documentation;

(3) change, and the slowness with which all its ramifications are taken into account, which leads to numerous versions of documents in various stages of evolution;

(4) lack of an end-to-end RT process, plus the absence of a specified RT job description, thus leading to RT mismanagement

(5)involvement of too many, often uncooperative people, with inadequate expertise and individual agendas.

Diverse Sources: literature

(1) Project longevity

(2) Lack of commitment by all parties

(3) Information complexity and hidden information

(4) Coarse granularity of traceable entities

(5) Immature integration technology

implications of diverse sources

lack of coverage

lack of focus

Diverse & Conflicting Needs: empirical study

- The ability to establish control by managing multiple copies of documents and any iterations, changes, additions, deletions, and so forth.
- The ability to enable the business case to drive the RE process so that each of its critical aspects is reflected in both the evolving RS and the end product.
- The ability to provide selective and filtered views of interconnected documentation to support different activities.

Diverse & Conflicting Needs: empirical study

- The ability to provide access to original requirements and all their stakeholders, as well as to further highlight where intermediaries have been involved, and so improve both information sharing and the communication potential.
- The ability to identify duplicated, conflicting, or non-verified requirements, then manage the transfer of these requirements between the different lifecycle phases.

Diverse & Conflicting Needs: literature

- The ability to promote a contractual approach to the development process.
- The ability to understand systems from multiple points of view and to assist in the pulling together of fragmented information
- The ability to offer some degree of assurance that specifications were written with user requirements in mind and so assist with user acceptance testing

Diverse & Conflicting Needs: literature

- The ability to track requirements allocation, requirements flow-down between development phases, and the rationale and constraints used to develop product elements. This also includes support for the analysis of aspects such as consistency, completeness, test procedures, data integrity, safety, security, and change impact

implications of diverse needs
compound problems
expectation management

Support for Post-RT

- Post-RT depends on the ability to trace requirements from and back to a relatively static baseline document, usually the RS, and through a succession of documents and products in which they are distributed. Most of the existing support for RT, particularly from the commercial tools, is directed at providing post-RT.
- Any further improvements with post-RT will only have a limited impact on reducing RT problems. This is because this type of RT does not reflect the fact that the baseline RS from which it operates is often only the end product of an on-going and exploratory process from which the requirements placed into this baseline emerge.

Support for Pre-RT

- Pre-RT is poorly understood and not comprehensively supported. Pre-RT depends on the ability to trace requirements from and back to their originating statement(s) and through their production and refinement process.
- Existing commercial tool support is not directly applicable for providing pre-RT. This is because: they generally treat the RS or equivalent baseline document as a black-box and provide little to show that the requirements are only the end product of a complex process; they tend to predefine rigid information categories for recording potentially traceable information and prematurely bind to requirements.

Impact

- Why Post-RT can only have limited impact on quality
  - the assumption that the requirements in the RS are relatively easy to obtain, accurate, and stable
  - the emergent nature of requirements
- Why Pre-RT might have a significant impact
  - quality culture from project inception
  - reduction of rework effort in reconstructing requirements rationale
  - economic leverage

The Roots of RM Failure

(1) A lack of shared or project-wide commitment, with no visibility of ownership and a lack of accountability, with the phrase "not invented here" being very common.

(2) Little cross involvement in work, and localised views of information, thus making it difficult to pin down the overall state of work or knowledge.

(3) Poor communication and distribution of information amongst teams, leading to much information loss, as well as the development of cliques over time.

(4) Changing notions of ownership and responsibility, due to continually changing work structures, and due to the turn-over of team members.

The Roots of RM Success

- (1) Clear visibility of participant responsibilities and knowledge areas.
- (2) Clarity of working structures and working relations.
- (3) Individuals who acted as common threads of involvement throughout the project and across project boundaries.
- (4) A strong sense of team commitment, accompanied by joint ownership.

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.
- Again, many of the problems here do not necessarily have technical solutions, and these will remain unless Requirements Management receives dedicated project resourcing.

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!

Action Plan [1]

management context
contract & procurement

- The lesson of the CMMI Software process improvements are interlocking. If you don't have "commitment control", don't waste your time and money on requirements management!
- Contract and procurement procedures are about managing the relationships between customer and supplier. The work of requirements management takes place in the "space" between customer and supplier. It may be necessary to "reengineer" contract and procurement procedures to: promote relationships based on partnership and risk sharing; support continuing and direct interaction between customer and supplier.

Action Plan [2]

individual performance

- There is good empirical evidence for the existence of striking differences in programming ability, similar results are reported for software design. There is strong anecdotal evidence of significant differences in individual performance at requirements engineering tasks. The most important step we can take in order to improve performance at requirements engineering is to select the right people. The key appears to be personal communication and group facilitation skills, generally accompanied by a sensitive appreciation of organisational politics.

Action Plan [3]

organisational setting

- Requirements management takes place in different organisational settings (internal, bespoke, customisation, cooperative, generic/market). The problems of requirements management are different in each of these settings. You cannot expect to use methods and processes from one setting in another without significant adaptation. When the setting changes you need to change your requirements management practices.

Action Plan [4]

bounding

- Establishing the scope and delineating the bounds of the requirements and design spaces is the most difficult and critical part of requirements engineering. Decisions on bounding must be explicit and clearly rationalised. They form the root for requirements management. The following tests should be applied and documented: investigation resources; competence; freedom of action; missed solution cost; robustness.

Action Plan [5]

make vs buy

- Software engineers have a habit of building rather than purchasing. Generally purchasing means a cheaper, more generic and hence more robust product. An assessment of likelihood of make vs buy needs to be made early in the requirements engineering process it should include analysis of products, patents and technical intelligence, papers and reports. An appropriate strategy for managing requirements when buying needs to be devised.

Action Plan [6]

requirements construction
collaborative work

- We tend to talk about requirements "capture", "acquistion", "elicitation". This 'butterfly net model' gives a false picture of requirements processes, requirements are not "out there". It is better to think of requirements "construction" a mutual exploration and learning process in which what is wanted is informed by what is possible. Requirements management needs to be sensitive to this process
- Requirements management is "collaborative work". There are lots of simple general purpose tools which can assist in this work - use them!

Action Plan [7]

stakeholder identification
information gathering

- The most common error in requirements engineering is to forget/omit/lose important stakeholders. It is important to create a "map" as a means of identifying stakeholders and interpreting the information provided by stakeholders and its status. This map should identify responsibilities, capacities and organisational relations.
- Gathering information on the requirements and on the domain in which they are situated is the characteristic activity of requirements engineering. There are better ways of gathering requirements than simply asking for them.

Action Plan [8]

modelling

- In order to render "raw" requirements usable they need to be organised through a process of modelling and specification. We need to produce 3 types of model: a system model which identifies the services the system is to provide and the assumptions that have to be made about the operational domain in order to provide those services; a task model which identifies the users and the tasks the users perform; a value model which identifies those properties of the service relevant to the fulfilment of the stakeholder requirements. To build these models we need to be able to talk about agents, goals, events, actions, objects and preferences.

Action Plan [9]

validation
inspection
prototyping

- To support validation it is important to be able to generate multiple dynamic views of requirements information. The most difficult parts of validation are: organising and documenting feedback; (& knowing what sort of questions to ask.)
- Inspection works (removes errors as near source as possible hence reducing costs of rework)! Use it as part of your requirements management strategy.
- Prototyping and system simulation work (as means of exploring system requirements)! Use them, but keep in mind the known problems.

Action Plan [10]

metrics
estimation

- "You cannot control what you cannot measure." It is necessary to establish measures of the products and process of requirements management. There are plenty of simple product, process and resource metrics which can and should be applied.
- Deriving estimates of development cost, effort and schedule is part of requirements engineering. We can certainly do a lot better than a finger in the air. The products of this process form part of the information that must be managed.

Action Plan [11]

rationale

- The current RE process is artifact-oriented great emphasis is placed on the creation and tracking of "products" however more than 70% of software development costs are in maintenance and rework and half the effort in these activities is about *understanding* the process which lead to those products so as to make effective corrections and enhancements. In order to achieve this understanding you need to know what decisions were considered, assumptions made, alternatives posited. This information is rationale, it may be remembered but with time and staff turnover it soon gets lost. There are many schemes for recording rationale which are simple, proven and available.

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.

Concluding Remarks

- There are no silver bullets but there are many simple things that can be done to improve requirements management which will have a major impact on quality.
- Be systematic about capitalising on your own experience.
- Invest in improving RE, it pays off!
- We have the expertise to help.