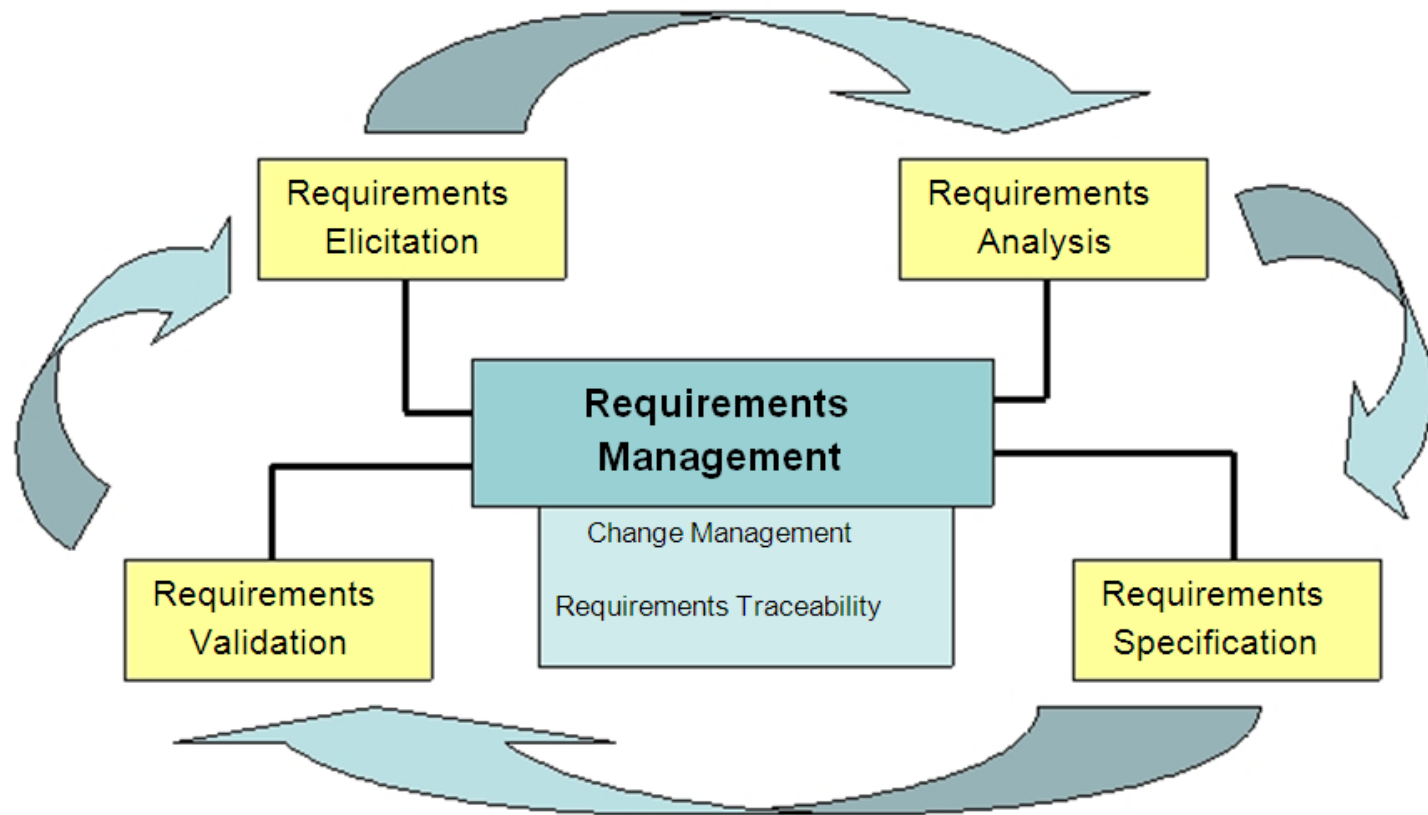


Requirement Engineering **Management**

References: C1.Ebook +John Vu (CMU)

Requirement Management



Change Management

- The process of managing change to the requirements for a system.
- The principal of requirements management:
 - Managing changes to agreed requirements.
 - Managing the relationships between requirements.
 - Managing the dependencies between the requirements document and other documents produced in the development process.

Stable & Volatile Requirements

- ❑ Requirements changes may occur while the requirements are being elicited, analyzed and validated and after the system has gone into service.
- ❑ Some requirements are subject to more 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.

Volatility

☐ Mutable requirements

- These are requirements which change because of changes to the environment in which the system is operating.

☐ Emergent requirements

- These are requirements that cannot be completely defined when the system is specified but which emerge as the system is designed and implemented.

Volatility

☐ Consequential requirements

- These are requirements which are based on assumptions of 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.

Activities (10 minutes)

- ☐ Identify:
 - Mutable requirements
 - Emergent requirements
 - Consequential requirements
 - Compatibility requirements
 - ☐ Build concept map
-

Change Factor

- ❑ Requirements errors, conflicts and inconsistencies:
 - As requirements are analyzed and implemented, errors and inconsistencies emerge and must be corrected.
 - Some of these may be discovered during requirements analysis and validation or later in the development process.
- ❑ Evolving stakeholders knowledge of the system:
 - As requirements are developed, customers and end-users develop a better understanding of what they really require from a system.

Change Factor

- Technical, schedule or cost problems:
 - Problems may be encountered when implementing a requirement.
 - It may be too expensive or take too long to implement certain requirements.
- 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.

Change Factor

☐ Environmental changes:

- The environment in which the system is to be installed may change, causing the system requirements to change in order to maintain compatibility.

☐ Organizational changes:

- The organization which intends to use the system may change its structure and processes, resulting in new system requirements.

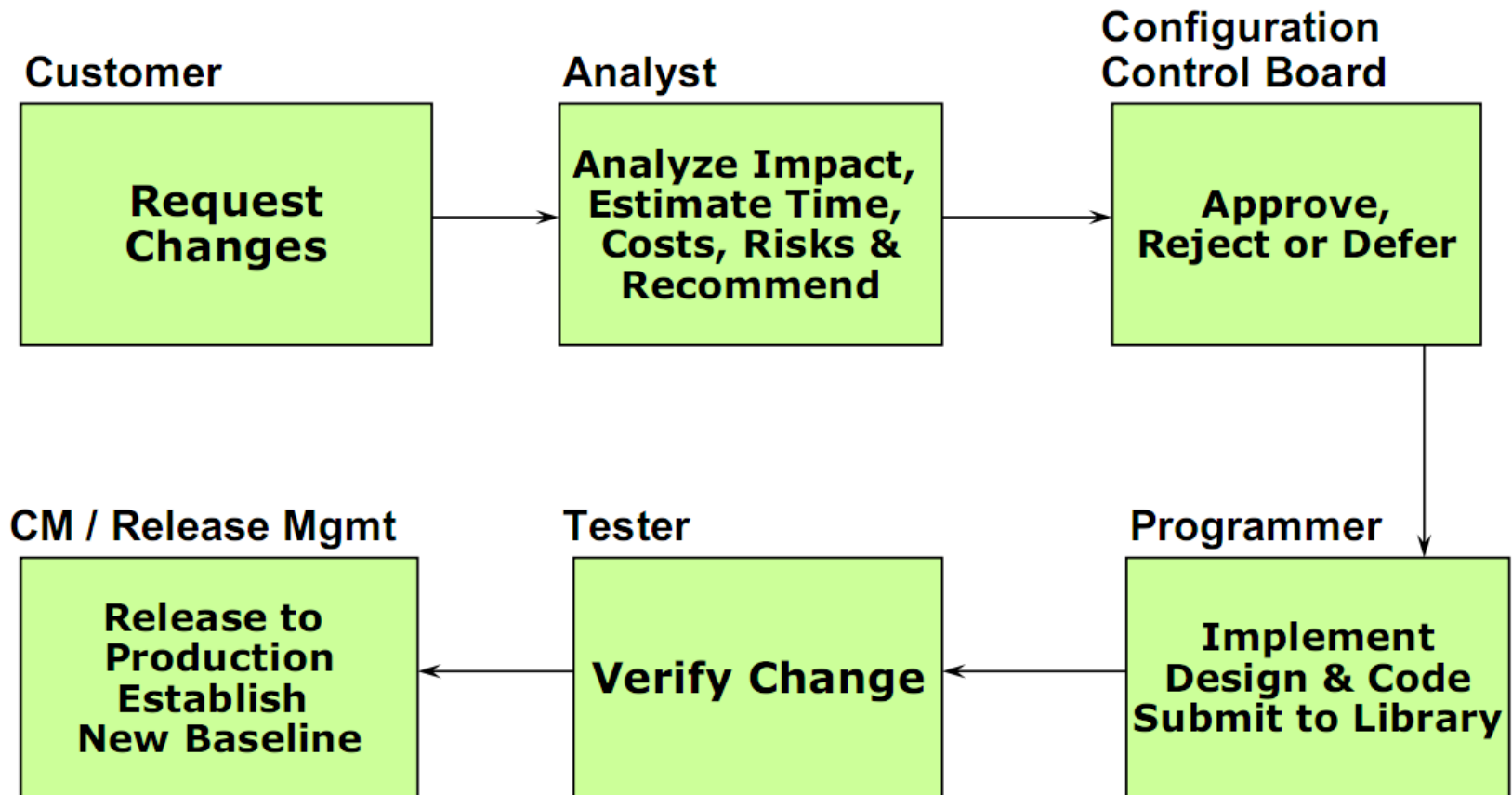
Change Management

- ❑ Change management are the **procedures, processes and standards** which are used to manage changes to requirements.
- ❑ Change management consists of:
 - 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.
 - The software support (if any) for the change control process.

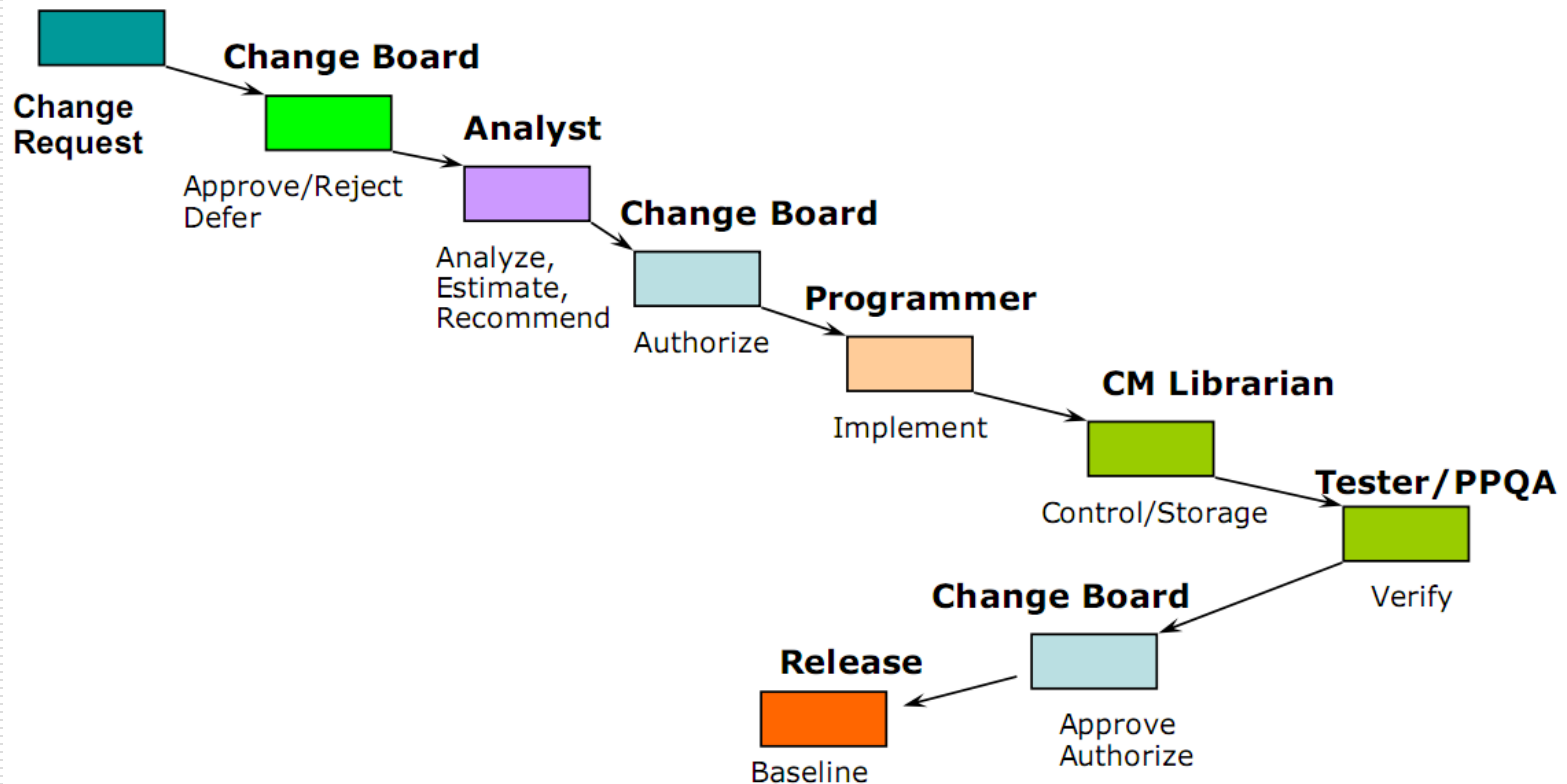
Change Management

- Change Management allows necessary changes to be made while ensuring that change impacts are understood project-wide.
 - Initial work for a product is done without change management.
 - The product is reviewed and baselined.
 - The baselined product is put under configuration management.
 - Further changes are treated systematically.
 - All changes are proposed via Change Board.
 - Analysts review changes, evaluate impact and make recommendations.
 - Change board prioritizes the change requests and accept, reject, or defer the changes.
 - Change board notifies all stakeholders of its decisions

Change Management Flow



Change Management Flow



Change Management Checklist

- ☐ Is the change request documented?
 - ☐ Is the change request analyzed?
 - ☐ Is the change request authorized?
 - ☐ Is version control current on the CIs?
 - ☐ Is the impact to other systems considered because of this change?
 - ☐ Is there traceability from change request to completed item?
-

Change Analysis

- ❑ 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 (CR) form which is then passed to all of the people involved in the analysis of the change.
 - ❑ Change request forms may include:
 - Proposed change
 - The change analysis
 - Data
 - Responsibility (Who is assigned)
 - Status field (Open/Close)
 - Comments field
-

Requirements Traceability

□ Purpose:

- To understand how **requirements changes impact other requirements** and downstream software development deliverables.
- Identifies **interdependencies** among requirements.
- Provides insight into the status of the development efforts by identifying what development deliverables exist to satisfy requirements.
- Demonstrates when requirements have been satisfied by associating them to system components and tests.

Traceability Benefits - 1

- Traceability provides a methodical and controlled process to manage the changes that occur as an application is developed.
- Without tracing **for every change, Software Engineers have to review every document** to see which elements of the project require updating.
- Without tracing, **it is costly, time consuming and difficult to establish whether all affected components** have been identified and updated.
- Without controlling all documents, changes can decrease system reliability over time.

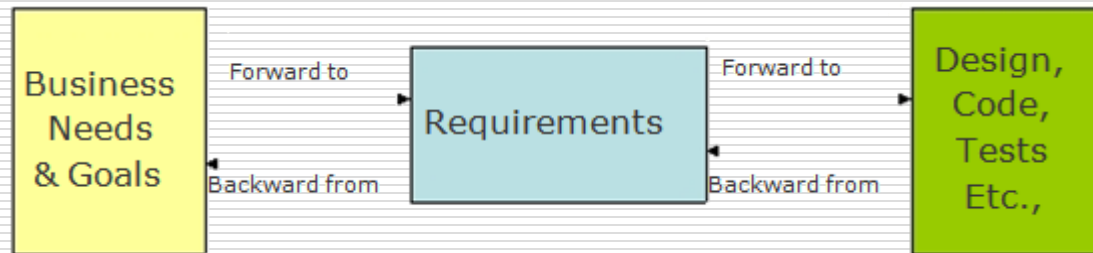
Traceability Benefits - 2

- With traceability, change management can proceed in an **orderly fashion**.
- The impact of a change can be understood by following the traceability relationships through the document hierarchy.
 - For example when the user needs changes, a developer can quickly identify which software elements must be changed, a tester can identify which test protocols must be revised, and managers can better determine the potential costs and the difficulty to implement the change.

Requirements Trace Matrices (RTM)

□ Definition:

- A requirements trace matrix (RTM) identifies how requirements are related to software development deliverables and to other requirements.
- Requirements matrices show related requirements and the forward and backward lineage to project deliverables.



Traceability

- ❑ Traceability information helps you assess the impact of requirements change. It links related requirements with 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 back to requirements.
 - *Forward-to traceability:* Links other documents (which may have preceded the requirements document) to relevant requirements.

Types of Traceability

- Requirements-source 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 (dependant and is-dependent upon).

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

| | 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 a 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

| Requirements | Depends On |
|--------------|------------|
| R1 | R3, R4 |
| R2 | R5, R6 |
| R3 | R4, R5 |
| R4 | R2 |
| R5 | R6 |

Traceability

- ❑ Traceability policies define what and how traceability information should be maintained.
- ❑ Traceability 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

- 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

□ Project team size and composition

- With a small team, it may be possible to assess the impact 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

Importance Of Traceability - 1

1) Verification and Validation

- Assessing adequacy of test suite
- Assessing conformance to requirements
- Assessing completeness, consistency
- Impact analysis
- Assessing over- and under- design
- Investigating high level behavior
- Impact on detailed specifications
- Detecting requirements conflicts
- Checking consistency of decision
- Making across the lifecycle

Importance Of Traceability - 2

2) Maintenance

- Assessing change requests
- Tracing design rationale

3) Document access

- Ability to find information quickly in large documents

4) Process visibility

- Ability to see how the software was developed
- Provides an audit trail

5) Management

- Change management
- Risk management
- Control of the development process

Traceability Difficulties

- 1) Cost
 - Very little automated support.
 - Full traceability is very expensive and time-consuming.
- 2) Delayed gratification
 - The people defining traceability are not people who benefit from it.
 - Development vs. Verification & Validation.
 - Much of the benefit comes late in the lifecycle.
 - Testing, integration, maintenance.
- 3) Size and diversity
 - Huge range of different document types, tools, decisions, responsibilities, but no common schema exists for classifying and cataloging these.
 - In practice, traceability concentrates only on baselined requirements.

Implement RTM

- The Requirements traceability matrix (RTM) can be implemented as a simple spreadsheet, with each row of the spreadsheet dedicated to a requirement to be implemented. Included are spreadsheet columns for the following:
 - A unique requirements identifier (REQ ID number).
 - A description of the requirement.
 - A pointer to a design document for the requirement.
 - A pointer to the unit test document for the requirement.
 - A pointer to the test results after testing the requirement.
 - A color-coded (i.e. red, yellow, or green) status column for the requirement.
 - A column for sign-off for the requirement's inclusion in the next release.

Implement RTM

- ❑ Each associated document, and all communication about the project (email, and so on) refer back to the requirements by requirement ID.
- ❑ This provides good two-way reconciliation between the RTM, and the set of design, test, and test-result documents.
- ❑ Meetings to measure project status and development progress always revolve around the RTM, and the requirements IDs contained within.
- ❑ This simple form of project status accounting helps to keep the project organized, and helps to focus all of the stakeholders.
- ❑ It also helps to ensure that nothing falls through the cracks.
- ❑ For instance, if a design document is missing for a particular requirement/feature, it will be obvious when you look at the spreadsheet. If the implementation of a particular requirement is not going well, the status column will show this (and you can sort them by status if needed).

Tools

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

RM Tools

- ❑ Change Management can be supported by requirements management tools or 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.

RM Tools

| Basic Function | DOORS | RDD-2000 | RTM | SLATE |
|---------------------|---|------------------------------------|--|------------------------------------|
| Database | N/A | Gemstone (OO) | Oracle (Relational) | <u>Versant ODBMS (OO)</u> |
| Public API | No | Yes | Yes | Yes |
| Modeling | Bridge | Embedded and Executable | Bridge | Bridge |
| Web enabled | No (html format output) | No (html format output) | Yes - fully interactive | In Development |
| User Extensible | Yes / basic schema <u>supplied</u> | Yes / basic schema <u>supplied</u> | Yes | Yes / basic schema <u>supplied</u> |
| Language | C++ with DXL (Doors Extension Language) | Small Talk | Sequel | <u>Tcl (Tool Control language)</u> |
| Focus | Document | Requirement | Document or Requirement | Document/Requirement |
| Defined Relations | Parent/Child | Named | Named | Parent/Child |
| Platforms | Unix/Windows/NT | Unix/Windows/NT | Unix/Windows/NT | Unix/Windows |
| Reporting Mechanism | Script - easy/not as <u>powerful</u> | Graphical - more robust | Script or Web form <u>creation</u> | Scripted |
| Publishing Tool IF | <u>Word, Ileaf, etc.</u> | Word, Ileaf | Word, Ileaf, <u>FrameMaker</u> | <u>FrameMaker</u> |
| Training/Support | <u>InHouse & Vendor</u> | <u>InHouse & Vendor</u> | Vendor Supplied - <u>inHouse</u> under Development | Vendor Supplied |

Benefits Of RM Tools

- ❑ Multi-disciplinary design data in one location:
 - Specifications can be generated and controlled.
 - Full requirements/functionality/verification traceability.
 - Facilitates integrated product team interaction.
- ❑ All data related through physical architecture and functionality to requirements:
 - Optimize design considering functionality, performance, cost trades.
- ❑ Facilitates quick impact analysis:
 - Requirement changes/traceability
 - Performance constraints
 - Physical constraints
 - Cost constraints
 - Total designs captures for reuse

Summary

- ❑ 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.