



Information Technology for European Advancement

Task 1.8 Requirements Management method D1.8.3

RM tools proposal

Version 02 - Public
Edited by Loek Bokhorst

Software Development Process for Real-Time Embedded Software Systems (DESS)

ITEA COMPETENCES involved:

- 1) Complex Systems Engineering**
- 2) Communications**
- 3) Distributed Information and services**

2001-12-10

ITEA

1. INTRODUCTION.....	3
1. REFERENCE.....	3
2. TOOLS REQUIREMENTS.....	3
2.1 Introduction.....	3
2.2 Summary of important tool requirements.....	4
2.3 Requirements elicitation and identification.....	4
2.4 Batch-mode document/source-link update.....	4
2.5 Supports components based development.....	5
2.5.1 Component re-usability.....	5
2.5.2 Reference requirements inheritance.....	5
2.5.3 Product line development.....	5
2.5.4 Import / export of components.....	5
2.5.5 The component name as part of the Traceability tag. E.g. UR-<component>. The "requirement type belongs to the User Requirements specifications.....	5
2.6 Traceability.....	5
2.6.1 Classification of links.....	5
2.6.2 Traceability at the component level.....	5
2.6.3 Default relations between components.....	5
2.6.4 Traceability to different object formats.....	6
2.7 Traceability analysis.....	6
2.7.1 Inconsistencies.....	6
2.7.2 Verification of requirement (was it done, how was done).....	6
2.7.3 Requirement of type constraints.....	6
2.8 Change Control.....	6
2.9 Configuration Management.....	7
2.9.1 Allow base-lining of requirements/set of requirements.....	7
2.9.2 Product-line support.....	7
2.9.3 Checking the consistency.....	7
2.9.4 Distributed repository support.....	7
2.9.5 Browsing through versions.....	7
2.9.6 Comparing different versions of requirements.....	7
2.9.7 Locking a single requirement.....	7
2.10 User Identification and access control.....	8
2.10.1 Intellectual property protection.....	8
2.11 Views.....	8
2.12 Documenting and Reporting Ability.....	8
2.12.1 Component level.....	8
2.12.2 Metrics.....	8
2.12.3 Extended Traceability views.....	9
2.13 Multi-level access control.....	9
2.14 Multi-site multi-user support.....	9
2.14.1 Web-based development support.....	9
2.15 Integration to other tools.....	10
2.16 System Requirements.....	10
2.17 Support for testability.....	10
2.18 Compliance to standards.....	11
2.19 Customizing ability.....	12
2.20 Quality.....	12
2.20.1 Portability.....	12
2.20.2 Recoverability.....	12
2.20.3 Stability.....	12
2.20.4 Extensibility.....	12
3. TOOLS OVERVIEW.....	13

1. INTRODUCTION.

This document lists the requirements for the ITEA DESS method based tooling for Requirements Management. The purpose is not to have a complete specification, but only those aspects of the tool are highlighted, which are important for this class of tools.

1. REFERENCE.

- [1] Evaluation Survey conducted by the International Council on Systems Engineering. (INCOSE). <http://www.incose.org/tools/tooltax.html>
- [2] **Requirements Engineering Tools** information provided by <http://www.systemsguild.com/GuildSite/Robb/retools.html>

2. TOOLS REQUIREMENTS.

2.1 Introduction.

An effective and tool-supported requirements management is a prerequisite for tackling the growing complexity of systems and the continuing pressure on reductions in development time and cost.

Effective requirements management is an important issue for the multi-site Development organisations. Increasing amounts of resources are allocated to requirements management with the aim of making their requirements more tractable. The main drivers for this effort are:

1. The movement to product families and platform-based software development;
2. An increased multi-site distributed development.
3. The growing product globalisation that results from the distributed nature of the stakeholders in a product family's set of requirements.

Furthermore, for efficient development there is a need for reuse of requirements. Multiple views on any given set of requirements must be available at any time. There is a growing need to be able to identify different subsets from a collection of requirements. For example: those that were specified for a given region, all common requirements for a given platform, all variations for a given feature.

"Multi-site multi-user, multi discipline" development, life-cycle tool chain integration, a high level of granularity are the important issues to be covered by the RM tool selected.

The basis of requirements management is the information needed and the accessibility of that by the respective types of requirements manager.

Some of the tools in the tool-chain of the development life cycle may already have the needed information available and accessible. The combination of traceability and requirements management information retrieval functions is the basis for an efficient process.

2.2 Summary of important tool requirements.

- Base-lining of components. From requirements till source code including traceability information and attributes. Browsing through versions.
- Security: Intellectual property protection.
- Component based development support. Import Export of components information
- Re-use of Components.
- Quality.
 - Database consistency checks.
 - Recovery.
- Extended Traceability views. Showing traces and attributes.
- Integration with other tools like configuration management, modelling, test, planning, E-mail, problem tracking.

2.3 Requirements elicitation and identification.

1. Unique numbers are assigned to a requirement.
2. Supports component based development.

The component name is part of the requirements identification.

3. Support for hierarchical numbering (e.g. 3.4.1) and lists (e.g. a., b., ... and (1), (2), ...) is to be provided.
4. The numbering style can be selected.
5. The requirements can be renumbered.
6. Standard attributes support.

It is possible to define default attributes per document type, requirement type.

These document types are:

- Requirements: commercial, user, system, hardware, software requirements
 - Architectural design,
 - Detailed design,
 - Validation and test: the acceptance test-, integration test-, component test-, white box test- specifications,
 - Source code.
7. Automatic creation of requirements based on keywords, grammatical structure, formatting or some identifiers, etc.(with a unique identifier given to each requirement)

2.4 Batch-mode document/source-link update.

Requirements that are updated, either directly or in batch operations, retain their links. New versions of documents may be used to update the requirements..

2.5 Supports components based development.

2.5.1 Component re-usability.

A component or package status information is accessible at the several abstraction levels: requirement-, architectural design-, design-, validation-, and source-code level. The respective tools in the chain will provide the information to the RM-tool.

2.5.2 Reference requirements inheritance.

The tool should support the user when inheriting components into the product under development.

2.5.3 Product line development.

A product configuration may use one or more variants of a component or sub-system.

2.5.4 Import / export of components.

From requirements till source code including up/down stream traceability information and attributes. Automatic import of electronic documents in various formats.

1. Export and synchronization of V&V modules and automatic traceability information.
2. Export, import and synchronization of project planning data.
3. Import and export of spreadsheets.
4. Output format: XML/XMI.
5. Import / export from/to other requirements management tools.

2.5.5 The component name as part of the Traceability tag. E.g. UR-`<component>`. The "requirement type belongs to the User Requirements specifications.

2.6 Traceability.

Visibility into existing links from source to implementation--i.e. follow the links.

2.6.1 Classification of links.

Types of links, links created for a special purpose other than requirements traceability can be defined and attributes can be connected.

2.6.2 Traceability at the component level.

An overview on component/product level can be made and printed.

2.6.3 Default relations between components.

The relations between information related to components can be defined. Components may have implicitly defined that each requirement of a certain type in the document has a "must be related to" a requirement defined in another document. E.g. each requirement in the CRS must have a relation to a requirement in the system requirements specification and the acceptance test specification.

2.6.4 Traceability to different object formats.

1. Traceability to information in UML diagrams at the lowest level e.g. timing constraints,
2. to information in graphical forms,
3. Traceability between components and single RM-objects and/or related information about the artefacts owned by the other tools used in the life-cycle.
4. The synchronization of the requirements with a planning tool,
5. Configuration management tool objects.
6. performance analysis tool,
7. static code analysis tools,
8. modelling tools,
9. Validation and Verification tools,
10. E-mail
11. Internet
12. to data of spreadsheet tools.
13. etc..

2.7 Traceability analysis.

2.7.1 Inconsistencies.

The tool should allow the user to identify inconsistencies such as unlinked requirements or system elements (orphans).

Visibility into existing links from source to implementation--i.e. follow the links

With the requirement links in place, the user needs the ability to follow the links to see where they come from and where they go to

2.7.2 Verification of requirement (was it done, how was done)

Throughout the life of the project, the requirement management tool will be used to verify that the requirements have been met. The tool should provide the ability to document that the requirement was fulfilled, how it was done, and who was responsible.

2.7.3 Requirement of type constraints.

Once constraints have been allocated to system elements, the requirements management tool should support the verification of those requirements by rolling up actuals and reporting on variances (this is the allocated weight versus the actual weight).

2.8 Change Control.

1. Access to all change request data and history is provided.
2. Change of the baseline version, change of the requirement and/or its attributes will automatically update the "requirement status" attribute.
3. The links to related object becomes "obscure".. See "Standard attributes support."
4. The tool shall support the history of change at the individual requirement level.

2.9 Configuration Management.

2.9.1 Allow base-lining of requirements/set of requirements.

1. Base-lining of requirements would thus prove helpful to reuse requirements across modules or even projects.
2. Per component the baseline can be defined. All links and the information(attributes) of the component are taken into account.

2.9.2 Product-line support.

1. Support variant development. Two or more branches of a single requirement set should be supported. Also merging of variant branches should be catered to.
2. Export and import of repository extracts on component basis.

2.9.3 Checking the consistency.

The traceability consistency check will guide the user to create a consistent product. A consistency check of the up- and down stream can be performed by specifying the list of document types. E.g. a consistency check between a requirement specification and the related test specifications

2.9.4 Distributed repository support.

At any level within the network of repositories all involved related configuration items are archived or restored automatically (e.g. without too much user-interaction).

2.9.5 Browsing through versions.

The integration of the RM tool allows implicit Configuration management actions to take place in order to access the documents, related information, like discussions, problem reports, change requests, design- and test-specification and results. All linked information and attributes of previous versions are restored.

2.9.6 Comparing different versions of requirements.

It shall be possible to compare different versions of the source/input and update without re-establishing any links. If a different version of a set of requirements is parsed, the tool should be able to identify or detect the difference and subsequently allow the requirements to be changed.

2.9.7 Locking a single requirement.

Multiple users are allowed to edit the same document. A single requirement can be locked by a user.

2.10 User Identification and access control.

The tool supports the definition of user types and user groups. The administrator is able to define per document the access and view rights per document. A distinction can be made between classified information both texts and attributes per user.

2.10.1 Intellectual property protection.

This information, including attributes, may be defined as hidden or not per user of the tool. The traceability to hidden objects is supported.

2.11 Views.

Default views and page layouts are available per document type and per type of user: management, project leader, architect, design-engineer, test-engineer, configuration manager.

Per type of organisation, for each type of user a list of attributes can be defined.

A list of documents can be made accessible per discipline or organisation:

The Commercial department, the Project management, the HW/SW development groups, the Quality department etc.

2.12 Documenting and Reporting Ability.

2.12.1 Component level.

An overview on component/product level can be made and printed. This information includes:

- Names of the components.
- Component size (code and memory)
- Component quality: requirements quality achieved.
- Component baseline information.
- Etc.

2.12.2 Metrics.

The tool supports a query facility in order to provide metrics information like:

1. Change activity.
2. Validation and test results, statistics like code coverage, Requirements quality.
3. Change activity: per type of specification, per project phase.
4. General statistics like Number of objects per repository, or per document.
5. Source code metrics.
 - Coding standard compliance information.
 - Complexity
 - Quality etc.

2.12.3 Extended Traceability views.

Traceability views, which show both the links of a specified type between the traceability items (TI) and some selected attributes. See example below.

CRS Req	CRS	FRS Req	FRS Req	SAD	SAD	SRS Req	SRS Req	Testgoal	Testgoal	Priority	Test Goal	Test Goal	Result	Remarks
Name field	Req. ID	Name field	ID	Req. ID	Req. ID	Name field	ID	Text	ID	Check box	Check box			
SW IP comp	CRS1	The demands	FRS1			The IDE Dr	SRSA1	AAA interface	SUTGA1	1	Interface ava	SUTGCA1	✓	This is a remark
SW IP RISC	CRS1	The demands	FRS1			Communic	SRSA2	Verify interfa	SUTGA2	1	for UOVA	SUTGCA2	✓	
SW IP RISC	CRS1	The demands	FRS3			Communic	SRSA2	Verify that th	SUTGA3	2	Communicat	SUTGCA3	✓	
SW IP RISC	CRS1	The demands	FRS4			Must supps	SRSA3	Verify that a	SUTGA4	1	Manual insp	SUTGCA4	✓	
Support for A	CRS2	The demands	FRS1					Verify that th	SATG1	1	Connect DVI	SATGC1	✓	
Support for A	CRS2	The demands	FRS3					Verify that th	SATG2	1	Interface ava	SATGC2	✓	
Support for A	CRS2	All the data m	FRS4			Must supps	SRSA4	Verify all HDI	SATG2	1	Threshold X'	SATGC3	✓	
Support for A	CRS2					Must supps	SRSA4	Verify all HDI	SUTGA5	1	HDD initialz	SUTGCA5	✓	This testgoal is not met
Support for A	CRS2					Must supps	SRSA5	Number of ss	SUTGA6	1	HDD initialz	SUTGCA7	✓	
Use linux	CRS3							Bla bla	SUTGA6.1	1			✓	
Use of board	CRS3.1					Will not sup	SRSA6	Verify that ss	SUTGA7	1	Int failure w	SUTGCA8	✓	
Reuse comp	CRS3.2							Bla bla	SUTGA7.1	1			✓	
									SUTGA8					

The view combines the some selected attribute values of a TI , the TI-name or the first part of the TI text field. with the traceability item identifier.

The view also shows the isolated, TI's which do not have a trace-in and or trace-out.

2.13 Multi-level access control.

1. Supports concurrent viewing, editing and management of requirements, at the same time taking care of security.
2. Support offline requirement viewing/editing and merging offline edited documents/requirements for individuals and groups based on specific access controls.

2.14 Multi-site multi-user support.

The accessibility and information retrieval functions are available to the various users types.

2.14.1 Web-based development support.

This would help the customers (individuals or groups) to have an insight into the status of the requirements. The tool should be able to take care of the necessary security aspects.

1. The tool should have the ability to support publishing of a limited set of requirements.
2. E-mails can be directly connected to a specific document or to a specific object defined in the requirements management repository. The E-mail history can be accessed.
3. Change requests and problem reports can be submitted. They can be directly connected to a specific document or to a specific object defined in the requirements management repository.

4. Access to the change request/problem report repository is supported.

2.15 Integration to other tools.

Tool	Related RM information
Configuration management tools. like Continuous, Clearcase.	Component label, component related documents, Development history, baseline (release) information
Design tools. Like Rational Rose, Telelogic Tau, Together, MESTRA, Rhapsody, Esterel Studio.	Graphical data etc., UML Component constraints information: memory usage etc.
Project planning/tracking tools. Like ms-project	Development costs Planning,
Validation and testing (management) tools	Realised quality (of a system, component, and requirement). test reports, test cases,
Problem tracking tools.	Problem reports / change requests Status of problem reports / change requests
E-mail tools. like Lotus Notes, ms outlook	Any correspondence on the requirements related to RM objects
Other tools like MatLab	Graphical presentation
Code Metrics tools.	<u>Project repository</u> : code sizes, etc. <u>Change requests</u> per requirement, document, component etc. <u>source code metrics</u> : complexity, maintainability, standards compliance

2.16 System Requirements.

Support multiple platforms. Ability to support mainly HP-UX (11.0), WindowsNT4 Workstation\Server, Windows2000 Professional\Server, Sun (2.6), Linux.

2.17 Support for testability.

1. Multi platform support. The test results per platform are made available for quality analysis.
2. Requirements Based Testing support.

2.18 Compliance to standards.

The tool complies with or supports standard templates for 'requirements writing'? (like the ISO/Military standards).

2.19 Customizing ability.

1. Support of state-of-the-art Repository Management Systems.
2. Repository Management System openness. Openness of the repository management system underneath the requirements management tool is needed for the storage and retrieval of requirements management information not covered by the respective tools in the life-cycle tool-chain. The DBMS enables the definition of the schemas and has functions available to the RM scripts for the repository maintenance and reporting functions.

2.20 Quality.

2.20.1 Portability.

The RM-tool runs on the following development platforms.

- HP-UX ,
- PC, Windows (NT etc).

2.20.2 Recoverability.

- A recovery file is created. After a crash a consistent version of the information is restored.
- The undoing of the changes made is available until the last checkpoint.

2.20.3 Stability.

The tool should be robust. Crashes of the tool should not occur. If exceptionally a crash does occur, recovery method should be quick and easy.

The tool should be able to recover from a crash with minimal loss of data and disruption to work.

2.20.4 Extensibility.

The RM tool can be easily extended e.g. by defining scripts for:
Attribute value consistency check, analysis.

3. TOOLS OVERVIEW

The answer for most of the questions will be given in a numeric scale that runs from 1 to 3. The minimum value (1) implies that the tool does not provide any form of support for the functionality referred to by the question or it only incorporates a very basic capacity, 2 indicates that it is partially supported, and a value of 3 indicates total support.

Criteria	DOORS (Telelogic)	RequisitePro (Rational)	RTM (Integrated Chipware)	Caliber RM
Interface to modeling tools	2: MATLAB-Simulink, ObjectGeode, Rational Rose, Stp, CoolJex, Many interface vendors, and interfaces provided by Telelogic. Granularity: ?	1: Rational Rose and other Rational tools, Note1: Requisite Extensibility may have improved this Granularity: :ReqPro could only link to a whole UML diagram	3: Rational Rose, Microsoft Word Interleaf Framemaker - Adobe Software through Pictures - Aonix Teamwork - Cadre Rose - Rational Soda - Rational Statemate-i-Logix Foresight-Nu Thena DocEXPRESS-ATA Paradigm Plus-Protosoft GEODE/Lov-Verilog : Very flexible interface. Granularity: to parts of the UML diagram	Via icon activation only down-stream at the MDL file level
Tools required?	none	Ms-word?	none	none
Export facility	2: Sgml, SQL, spreadsheet Framemaker, ms-office, RTF, HTML,OLE, Interleaf	2: Excel, word	3:Very flexible interface	Same as DOORS
Import Facility	2: plain text files, RTF, spreadsheet, mif, ms-office files ,	2: Excel, word	3: Very flexible interface	Import of a DOORS requirements repository done in pactice. Rest: same as DOORS
Configuration management interface: abstraction level unknown	Continuus, Clearcase, PVCS	Default to Clearcase, Continuus being implemented	To be checked	Default to PVCS, A SSC API interface is defined.
Graphical view doc. hierarchy	Yes			
Presentation of matrices	poor	good		good, also export of data possible
Learning curve	3	2	1: very flexible, so hard to learn	2
Metrics generation	2: due to export functions to EXCEL and SQL, the availability of DXL script language	2	2: Very flexible interface	Standard form for nr. of changes per requirement
Support test process	1: Links to test case, test results	1	2: Testexport, Very flexible interface, TestDirector	1: Caliber-RBT(Requirement Based Testing)., other tools via icon activation down-stream only
Documentation	3	3	1: is being improved	to be checked.
Performance	2	2	2	ODBMS should perform better than a relational DBMS.
Impact analysis	3 Implicit feature	2 By creating special views	3: Very flexible interface	Only via icon activation downstream to other

ITEA

Criteria	DOORS (Telelogic)	RequisitePro (Rational)	RTM (Integrated Chipware)	Caliber RM
				tools.
traceability analysis	3 Implicit feature	2 By creating special views	3: Very flexible interface	Implicit feature
Document editing	editing single objects	Editing via Ms-word(constraint)	IcFrame, IcWord	
UI(speed of interaction)	1: version 5 should be according to the windows standard	3: good, windows standard	2:	3:good, windows standard
Pre-defined Templates for UML,		yes		
Prices: 1	Floating or node locked: □ 6,354, Maintenance: □ 953 / year	node locked □ 1,909, floating: □ 3,818 (3 users) Maintenance □323	Life contract: □8,146 Maintenance 15%	Seat Licence: □2,068,= Concurrent(4 users) : □6,619,= server : □16,549 (windows NT) Maintenance 15%
Training for Customising	DOORS: 3 days DLL programmer 1 day	1 day: On the job:	administrator: 1 day,	
Training normal users	2 days		3 days	5 days installation, including 3 day training
Platforms	Windows 95,98,NT, Polaris, HP/USX, AX and others	Windows 95,98,NT	SUN Spar-SunOS (BSI) & Polaris (SFr4) HP-HP-USX IBM RS6000-AX DEC-Digital UNIX SIG-IRIS PC-Windows 95 & Windows NT	
project size	middle to large	small to middle	large	large
repository	Edit Texts and links using repository	Maintain links using an integrated repository		Maintain links using an integrated repository: Object Oriented DBMS: Versant ODBMS.
Openness of the repository management system: DBMS schema (and definition of it) functions available to the RAM scripts	?		Yes OK++(DBMS and open schema)	Via forms definable DB record layout definitions. OK++ (ODBMS)

Tools URL's.

<http://www.incose.org/tools/tooltax.html>

<http://www.atlsysguild.com/GuildSite/Robs/retools.html>