TECHNICAL REPORT

ISO/IEC TR 24766

First edition
2009-12-15

Information technology — Systems and software engineering — Guide for requirements engineering tool capabilities

Technologies de l'information — Ingénierie des systèmes et du logiciel — Guide pour les capacités d'outil d'ingénierie des exigences

Reference number
ISO/IEC TR 24766:2009(E)

ISO copyright office
Case postale 56 • CH-1211 Geneva 20
Tel. + 41 22 749 01 11
Fax + 41 22 749 09 47
E-mail copyright@iso.org
Web www.iso.org

Published in Switzerland

Contents

Page

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work. In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1.

International Standards are drafted in accordance with the rules given in the ISO/IEC Directives, Part 2.

The main task of the joint technical committee is to prepare International Standards. Draft International Standards adopted by the joint technical committee are circulated to national bodies for voting. Publication as an International Standard requires approval by at least 75 % of the national bodies casting a vote.

In exceptional circumstances, the joint technical committee may propose the publication of a Technical Report of one of the following types:

— type 1, when the required support cannot be obtained for the publication of an International Standard, despite repeated efforts;

— type 2, when the subject is still under technical development or where for any other reason there is the future but not immediate possibility of an agreement on an International Standard;

— type 3, when the joint technical committee has collected data of a different kind from that which is normally published as an International Standard ("state of the art", for example).

Technical Reports of types 1 and 2 are subject to review within three years of publication, to decide whether they can be transformed into International Standards. Technical Reports of type 3 do not necessarily have to be reviewed until the data they provide are considered to be no longer valid or useful.

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO and IEC shall not be held responsible for identifying any or all such patent rights.

ISO/IEC TR 24766, which is a Technical Report of type 2, was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 7, *Software and systems engineering*.

Introduction

Requirements engineering (RE) is a major activity within the systems and software engineering life cycles. This activity must be carried out in a comprehensive manner to ensure that a complete set of user needs and requirements is captured. These user needs and requirements are transformed into a validated set of technical requirements and managed throughout the life cycle using the RE process activities. RE tools are used to support many RE and related life cycle activities. RE processes are identified in ISO/IEC 15288:2008, *Systems and software engineering — System life cycle processes* and ISO/IEC 12207:2008, *Systems and software engineering — Software life cycle processes*.

ISO/IEC 15288:2008 and ISO/IEC 12207:2008 describe a set of RE processes, activities and tasks to be performed when acquiring or developing systems and software. However, these documents do not address the RE tool capabilities users can expect in order to support an RE process and other related life cycle activities.

Many RE processes are human activities that, in the current state of the practice, tools cannot perform, and that might never be able to be performed by a tool. But wherever possible, a tool should support these human activities through the facilitation of documentation capture, content management, distribution, discussion forums, and decision support tools.

This Technical Report describes capabilities of RE tools to benefit the groups of people that acquire, supply, develop, operate, and maintain an RE process.

This Technical Report will help RE personnel involved in the execution of one or more RE activities to

- obtain a better understanding of the relationship between the activities in which they are involved and RE tool capabilities,
- identify processes or activities that can be improved through better support by an RE tool, and
- have an objective basis for a better comparison, evaluation and assessment of RE tools.

This Technical Report will help people involved in the purchase of RE tools to

- review RE services that can contribute to RE process improvement, and
- identify criteria for selecting RE tools.

This Technical Report will help RE tool vendors to

- provide RE tools consistent with ISO/IEC 15288:2008, ISO/IEC 12207:2008, ISO/IEC 15940:2006, and ISO/IEC 14102:2008.

Information technology — Systems and software engineering — Guide for requirements engineering tool capabilities

1 Scope

Requirements engineering (RE) is an essential process of the systems and software engineering life cycles. RE has been established as an ISO/IEC standard life cycle process in both ISO/IEC 15288:2008, *Systems and software engineering — System life cycle processes* and ISO/IEC 12207:2008, *Systems and software engineering — Software life cycle processes*.

This Technical Report provides guidance on desirable capabilities of RE tools. It supplements ISO/IEC 14102:2008, *Information technology — Guideline for the evaluation and selection of CASE tools*, which details a set of evaluation criteria for CASE tools without referencing a specific activity or service area.

2 Normative references

The following referenced documents are indispensable for the application of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

ISO/IEC 12207:2008, *Systems and software engineering — Software life cycle processes*

ISO/IEC 15288:2008, *Systems and software engineering — System life cycle processes*

3 Terms and definitions

For the purposes of this document, the following terms and definitions apply.

3.1
activity
set of actions that consume time and resources and whose performance is necessary to achieve, or contribute to, the realization of one or more outcomes

3.2
manage
provide storing and editing capabilities, tracking history of edition, versioning, author identification, change management, time stamping, user notification for content changes, security rights control

3.3
management
provision of storing and editing capabilities, tracking history of edition, versioning, author identification, change management, time stamping, user notification for content changes, security rights control

3.4
functional requirement
requirement that specifies a function that a system or system component must be able to perform

[ISO/IEC 25000:2005]

3.5
quality requirement
non-functional requirement
capability of a product to satisfy the stated and implied needs when used under specific conditions

3.6
process
set of interrelated or interacting activities which transforms inputs into outputs

[ISO/IEC 12207:2008 and ISO/IEC 15288:2008]

3.7
requirements attributes
set of properties associated with requirements

3.8
stakeholder
party having a right, share, or claim in a system or in its possession of characteristics that meet that party's needs and expectations

[ISO/IEC 25000:2005]

3.9
stakeholder equity
degree of the share or claim a stakeholder has in the system of interest or a portion of the system of interest

3.10
user requirements
expression of perceived need from individual or group that benefits from a system during its utilization

NOTE User requirements are requirements issued by a user.

4 Requirements engineering process

4.1 Overview

A requirements engineering (RE) tool should facilitate and support the systematic managing of requirements throughout the project life cycle. The tool should also support the related activities in the context of the RE process. The following sub-topics characterize the processes that an RE tool would need to address.

4.1.1 Requirements elicitation

Requirements elicitation is the process of seeking, uncovering, acquiring, and elaborating requirements. Requirements are elicited rather than just captured or collected. This implies there are discovery, emergence, and development elements to the elicitation process. Requirements elicitation is an iterative activity and benefits from continuous communication and validation with stakeholders.

4.1.2 Requirements analysis

Requirements analysis involves refining the requirements by decomposing high level requirements into details, building prototypes, evaluating feasibility, analyzing overlaps or conflicts between requirements, and negotiating priorities. The goal is to develop requirements of sufficient quality and detail to reflect the stakeholders' needs.

4.1.3 Requirements specifications

Requirements specification deals with documenting the requirements in a consistent and reviewable way. Documentation includes the functions and capabilities that a system must provide and the constraints that a system must respect. Requirements specification is the basis for all subsequent software and system life cycle activities including project planning, design, and production, as well as the foundation for system testing and user documentation.

4.1.4 Requirements and product validation

Validation is the process of evaluating a system or software to determine whether it meets stakeholder requirements. It is performed by examination and through the provision of objective evidence that the requirements for a specific intended use or application have been fulfilled.

4.1.5 Requirements and product verification

Verification is the process of evaluating a system or software to determine whether it properly reflects the specified requirements. It is performed by examination and through the provision of objective evidence that specified requirements have been fulfilled.

4.1.6 Requirements management

Requirements management in conjunction with change management ensures that the requirements remain aligned with the developed product. Requirements management concerns the collection, analysis, and validation and verification of requirements with all the communications and negotiations inherent in working with people.

5 Requirements engineering tool capabilities

5.1 Overview

RE tools allow requirements engineering and management actions to be automated, reducing the cognitive load on the stakeholder. This section provides a list of required capabilities for an RE tool. The capabilities are organized according to the system and software requirements activities listed in ISO/IEC 12207:2008, ISO/IEC 15288:2008, and ISO/IEC TR 19759 (SWEBOK).

This list can be used for:

- Evaluating and choosing an RE tool
- Matching process to specific RE tool capabilities

5.2 Requirements elicitation

5.2.1 Overview

The requirements expressed in the project scope must address all essential business and user needs. The RE tool should be able to support in identifying stakeholder, capturing and tracing of the business/user requirements, functional requirements, and the quality requirements during elicitation work.

5.2.2 Requirements capture

The RE tool should support requirements capture by allowing the user to:

- Storing and managing the documentation from interviews, workshops, and observation

- Storing and managing stakeholder information (e.g., contact lists, comments, and etc.)
- Tracing requirements and generating trace reports
- Creating hierarchical relationships between requirements
- Including design rationale information directly associated with any hierarchical link
- Importing text and graphics from applications (e.g., open text formats)
- Updating existing linked documents from new or changed versions of the source documents without having to re-establish traceability links
- Storing and managing attributes for classifying or categorizing requirements during identification
- Storing and managing attributes in a variety of formats (e.g., text, enumerated, binary, graphics, descriptions, attachments), that can be associated with each requirements
- Using tool generated inherent attributes (e.g., unique requirement identification, author, time date, requirements change history)
- Using flexible search options for requirements by word or attributes (e.g., requirements identifier, words in text files, user and tool generated attributes)
- Managing the replacement or updating requirements by manual or electronic import
- Using flexible user programming language to develop reports for display or generating documents from the tool

5.2.3 Capturing “as-is” and “to-be” system elements

Requirements expressed in the project scope must not exclude any essential business, user, functional, and quality requirements. The RE tool should support capturing "as-is" and the "to-be system elements as follows:

- Storing and managing graphics and text (e.g., architecture, functional decomposition, Work Breakdown Structure (WBS))
- Storing and managing user definable attributes and additional information associated with a given requirement (e.g., stakeholders, business process, activity, tasks, policy, constraints)
- Flexible tracing (e.g. forward and backward tracing, one to many and many to one, bi-directional tracing of text to text, text to graphics, graphics to graphics, elements within graphics, tables and cells within a table)
- Bi-directional tracing of additional requirements and link between them (e.g., requirements to requirements, requirements to derived requirements
- Bi-directional tracing of requirements to system elements
- Bi-directional tracing the allocation of requirements to system elements
- Bi-directional tracing of rationale, assignments, criticality, test and validation to the requirements, allocation, and system elements

5.2.4 Stakeholder and requirements traceability

All functional and quality requirements should be traceable back to specific user, stakeholder, and business requirements. The RE tool should support traceability between them as follows:

- Storing and managing the identification and documentation of stakeholders and their roles and responsibilities
- Flexible searching and reporting of inconsistencies such as unlinked requirements or system elements (e.g., orphans)
- Bi-directional tracing of user needs and requirements
- Tracing user defined attributes for requirements that was fulfilled, how it was done, and who was responsible
- Displaying of traceability in graphical and textual form
- Flexible exporting of traceability matrix in both textual and graphical forms (e.g. Comma Separated Value (CSV), open text format, eXtensible Markup Language (XML), and etc..)

5.2.5 Goal-oriented scenarios and high-level modeling

Scenarios, models, and simulations can be used to describe the specific interaction between a user and a system to accomplish the goal of requirement. The RE tool should support goal-oriented scenarios and modeling as follows:

- Storing and managing user defined or tool provided templates for goal-oriented scenarios (e.g., simulations and modeling business scenarios, strategic issues)
- Storing and managing user defined scenarios
- Evaluating requirements based on business goals

5.2.6 Elicitation templates and checklists

Templates and checklists provide a consistent structure for recording the requirements descriptions and other requirements related information. The RE tool should support elicitation templates and checklists as follows:

- Storing and managing user defined or tool provided templates for elicitation (e.g. Quality Function Deployment (QFD) or Goal Question Metric (GQM))
- Storing and managing user defined or tool provided elicitation checklists
- Storing and managing user defined or tool provided prioritization forms

5.2.7 Prototyping

Prototyping can be used to explore and validate requirements. The RE tool should support prototyping as follows:

- Presenting information in a graphical user interface (GUI)

5.2.8 Importing and exporting to and from other sources

Requirements should be imported from, or interfaced to users, hardware, and other software systems. The RE tool should support Importing and exporting to and from other sources as follows:

- Importing and exporting to and from other tools (e.g. verification, design, spreadsheets, project management, documents)
- Importing and exporting to and from various standard file formats (e.g., Comma Separated Value (CSV), eXtensible Markup Language (XML))

5.2.9 Elicitation documentation

The output from the entire requirements elicitation tasks should be stored, retrieved, and edited in various formats. The RE tool should support elicitation documentation as follows:

- Storing and managing non-textual requirements in the specified format (e.g., bit-mapped graphics, vector graphics, tables, equations, or formal logic notations)
- Storing and managing textual requirements statements using basic text processor and spell checker

5.3 Requirements analysis

5.3.1 Overview

Requirements analysis includes decomposing high-level requirements into details by building prototypes, evaluating feasibility, and negotiating priorities. The RE tool should be able to support in decomposing requirements into functional and quality requirements, and in analyzing requirements feasibility and risk.

5.3.2 Functional requirements analysis

Functional requirements are a statement of required functionality or a behavior that a system will exhibit under specific conditions. The RE tool should support functional requirements analysis as follows:

- Storing and bi-directional tracing of identified user requirements to functional requirements
- Hierarchical structuring and identification scheme for the elaboration of requirements
- Checking spells and grammars on requirements statements (e.g. word spelling check, passive vs. active voice)
- Bi-directional tracing of requirements analysis to system implementation (e.g., architecture design, Work Breakdown Structure (WBS))

5.3.3 Quality requirements analysis

Quality requirements analysis involves significant architectural and design decisions. The RE tool should support quality requirements analysis as follows:

- Storing and managing quality requirements in quality attributes, policies, or constraints
- Hierarchical structuring and identification scheme for the elaboration of requirements
- Bi-directional tracing of quality requirements to source requirements or user requirement
- Checking spells and grammars on requirements statements (e.g. word spelling check, passive vs. active voice)
- Bi-directional tracing of requirements analysis to system implementation (e.g., architecture design, Work Breakdown Structure (WBS))
- Storing and managing the results or the rationale of quality attributes trade-off

5.3.4 Feasibility analysis

Feasibility analysis evaluates the possibility of implementing each requirement at acceptable cost and performance. It also identifies technical obstacles. The RE tool should support feasibility analysis as follows:

- Storing and generating user defined or tool provided checklists or templates for various analysis, (e.g., technical, economical, and operational analysis)
- Storing and managing rationale of feasibility analysis

5.3.5 Modelling

Modeling analysis depicts the requirements at a high level of abstraction. Such models include data flow diagrams, entity relationship diagrams, or UML diagrams. The RE tool should support modeling analysis as follows:

- Importing and exporting to and from modeling tools and displaying the results
- Storing and displaying context diagrams, conceptual domain models, and other high level models (e.g. Goal models, Object models, Task models)
- Storing and displaying analysis of requirements in graphical form (e.g., Unified Modeling Language (UML), Data Flow Diagram (DFD))

5.3.6 Prototyping

When developers or users are not certain about the requirements, constructing a prototype make the concepts and possibilities more tangible. The RE tool should support prototyping as follows:

- Presenting information in a graphical user interface (GUI)
- Demonstrating algorithm

5.3.7 Attribute analysis

User defined attributes such as risk, priority, and cost provides metrics for tracking requirements based on project needs. These attributes are assignable to each requirement. The RE tool should support tracking of any attribute(s) user defined or tool provided as follows:

- Storing and managing attributes in various formats (e.g. text, numeric, graphics, attachments)
- Detecting and flagging missing attributes
- Storing, sorting, grouping and ordering of attributes
- Managing changes to attributes

5.3.8 Requirements refinement

Requirements refinement identifies the hierarchical relation of requirements and allocates the requirements to subsystems. The RE tool should support requirements refinement as follows:

- Hierarchical structuring and identification scheme for the elaboration of requirements
- Bi-directional tracing of requirements to parent requirements or user requirement
- Bi-directional tracing of requirements to child requirements or design elements

- Bi-directional tracing of requirements to verification and validation case element
- Storing and managing the allocation rationale (e.g., how it was done, who was responsible)
- Supporting a flexible set of user defined or tool provided queries on requirements

5.3.9 Risk analysis

Risk analysis provides a standard approach to identify and document potential risks, and propose strategies for mitigating them. The RE tool should support risk analysis as follows:

- Exchanging information pertinent to risk analysis tools with external risk analysis tools
- Associating requirements with risks as a means of mitigation

5.3.10 Decision methods

Multiple stakeholders decide how to manage and resolve conflicting requirements. The RE tool should support decision methods as follows:

- Supporting an interface to a possibly separate terminology(glossary) or domain knowledge repository
- Checking requirements' conformity to predefined templates
- Storing and managing the list of conflicting requirements
- Storing and managing the information and rationale for resolved results

5.3.11 Requirements analysis artifacts

The output from the entire requirements analysis task should be stored, retrieved, and edited in various formats. The RE tool should support requirements analysis documentation as follows:

- Storing requirements analysis results in both text and graphical formats (e.g., Rich Text Format (RTF), eXtensible Markup Language (XML),JPEG)
- Storing, managing, and exporting requirements analysis results using basic text processor and spell checker
- Storing and managing non-requirements items, such as constraints on resources (e.g., time, budget, people) and considerations on design

5.4 Requirements specification

5.4.1 Overview

The specification states the functions and capabilities that a software or a system must provide and the constraints that it must respect. The RE tool should be able to support for all requirements be specified and documented in a consistent, accessible, and reviewable way.

5.4.2 Developing specification

The specification should describe as completely as necessary the software or the system's behavior under various conditions. The RE tool should support developing specification as follows:

- Generating traceability reports the identify exceptions in user requirements in the analysis phases to functional requirements

- Generating reports that identifies the exceptions for each functional requirement
- Generating and exporting templates in a variety of standard formats (e.g., text, Rich Text Format (RTF), eXtensible Markup Language (XML))

5.4.3 Traceability analysis

During requirements analysis, changes must be documented. Traceability establishes and tracks the linkage between each requirement and its source, proving insight into the overall project status. The RE tool should support traceability analysis as follows:

- Generating bi-directional traceability of requirements to requirement sources or other artifacts
- Displaying a graphical representation of the requirements hierarchy
- Generating a report for missing and/or orphan requirements
- Generating a report that identifies changes from/to a requirements throughout the development life cycle
- Generating reports of traceability attributes (e.g., category, number of approval or unapproved, number of changes, number of pending)
- Generating and exporting traceability matrix tables
- Generating a status report for the status of each requirement based on user defined or tool provided attributes

5.4.4 Requirements specification documentation

The output from the entire requirements specification task should be stored, retrieved, and edited in various formats. The RE tool should support requirements specification documentation as follows:

- Exporting document in standard formats (e.g. Rich Text Format (RTF), MS-word)
- Generating user defined or tool provided templates. (e.g. military standard formats)
- Checking the document through spell checking, grammar checking, data dictionaries, and acronym tables
- Generating presentation-quality charts and graphs in standard formats (e.g. PIC, PNG, JPEG, GIF)
- Generating the output of the specification in a finished form, including: page security markings, graphics or figures, user definable tables, and indexes
- Synchronizing changes between the RE tool and formatted document (complete loop between RE tool and formatted document)

5.5 Requirements verification and validation

5.5.1 Overview

Requirements verification and validation ensure that the requirements are correct and that the stakeholders agree to the requirements. Verification and validation activities are not a single discrete phase but an iterative process performed during elicitation, analysis, and specification. The RE tool should be able to support the various tests and evaluation means in verifying and validating the requirements.

5.5.2 Review and inspection

Reviewing and inspecting requirements is used for identifying ambiguous requirements and for validating requirements. The RE tool should support review and inspection as follows:

- Accessing to large numbers of stakeholders and users for review and comment of requirements
- Providing user and group security for each requirement (e.g., access, read only, read-write)
- Generating reports on requirements by user defined sort criteria (e.g., attributes, key words)
- Managing user comments and review edit
- Storing and managing discussion threads linked to a specific requirement
- Maintaining history of all changes for each requirement
- Storing and managing the review and inspection results (e.g., how it was done and who was responsible)

5.5.3 Verification

Verification is a more formal and rigorous way of ensuring that the final product meets the specified requirements. The RE tool should support verification as follows:

- Tracing of verification cases to the requirements
- Generating exception reports on requirements that do not have verification plan cases and verification plan cases that are not linked to requirements
- Supporting review and inspection of verification plans and procedures
- Tracing of verification cases to the requirements
- Tracing of the verification cases to verification procedures
- Storing and managing verification plans and verification procedures
- Generating exception reports on verification plan cases that do not have verification procedures and verification procedures that are not linked to verification plan cases
- Providing standard format for interfacing to verification tools

5.5.4 Validation

Validation is a way of ensuring that the final product meets the users or stakeholder needs under expected usage conditions. The RE tool should support validation as follows:

- Tracing of validation cases to the user requirements
- Generating exception reports on user requirements that do not have validation plan cases and validation plan cases that are not linked to user requirements
- Supporting review and inspection of validation plans and procedures
- Tracing of validation cases to the user requirements

- Tracing of the validation cases to validation procedures
- Storing and managing validation plans and validation procedures
- Generating exception reports on validation plan cases that do not have validation procedures and validation procedures that are not linked to validation plan cases
- Providing standard format for interfacing to validation tools

5.5.5 Defining acceptance criteria

Acceptance verification and end to end validation evaluates whether the product satisfies its documented requirements (verification) and whether it is fit for use in the intended operating environment (validation). As part of the acceptance verification plan, the acceptance criteria must be established. The RE tool should support defining acceptance criteria as follows:

- Storing and managing the acceptance criteria
- Supporting review and inspection of acceptance criteria
- Generating exception reports on requirements that do not trace to verification acceptance criteria and verification acceptance criteria that do not trace to requirements
- Generating exception reports on goals that do not trace to acceptance criteria and acceptance criteria that do not trace to goals
- Generating exception reports on system interfaces or requirements that do not trace to standards and standards that do not trace to system interfaces or requirements
- Generating exception reports on requirements that do not trace to a detailed design element and detailed design elements that to not trace to a requirement
- Generating exception reports on detailed design elements that do not trace to an implementation element and an implementation element that is not linked to a design element

5.5.6 Baseline of the requirements

Once the requirements artifacts are in- process validated and approved by the user and/or stakeholders, each artifact is deemed a baseline. A baseline is the agreed upon and approved set of requirements. The RE tool should support baseline of the requirements as follows:

- Storing and managing a baseline document
- Tracing the baseline requirements to a specific product release (specification tree)
- Version controlling of baseline requirements
- Generating reports that compare and contrast baselines versions
- Providing read and write protection of baseline products

5.6 Requirements management

5.6.1 Overview

Requirements management is a life cycle activity (lasting beyond the development phase to retirement and disposal). The RE tool should be able to support for monitoring the changes and the maintenance of the requirements in insuring that the requirements accurately reflect the product.

5.6.2 Identification of configuration items of requirements

After the baseline requirements have been established, changes may be made only through the defined change process. Every version of the requirements needs to have a unique identifier to avoid confusion between drafts and baselines and between previous and current versions. The RE tool should support Identification of configuration items of requirements as follows:

- Managing version identification (e.g., version number, date, time of creation or revision)
- Searching and tracing of specific versions of requirements

5.6.3 Requirements change management

Change management tracks changes to requirements and ensures that approved changes are communicated to all affected stakeholders. The RE tool should support requirements change management as follows:

- Providing check in and out capabilities of the baseline to add, change, update requirements
- Providing user defined or tool provided flexible search criteria for all requirements artifacts affected by the changed requirements
- Managing the change status (e.g., acceptance or approval, for pending changes)
- Storing and generating electronic change request form and templates
- Accessing to and distributing electronic forms and electronic mail notification
- Generating requirements modification history reports that should include, change author, change id, date of change, change status such as accepted, rejected or pending

5.6.4 Traceability management

Traceability management allows one to follow the life of requirements both forward and backward, and from origin through implementation. The RE tool should support traceability management as follows:

- Storing and managing references and links to source documents
- Tracing requirements to other sources (e.g., contractor's requirements, design, and verification procedure documents)
- Generating reports that compare current and previous versions when a source document is updated
- Storing historic information on the requirements by various requirements attributes (e.g., time, owner)
- Maintaining a history of requirements changes, who changed it, when it was done, why it was done

5.6.5 Risk management

Risk management is an approach to identifying and documenting risk factors, evaluating their potential severity, and proposing strategies for mitigating them. Risk management relates to requirement engineering in two ways: Risks may be raised against requirements, and requirements may mitigate risks.

While it is not the role of the RE tool set to manage risks, it is important that the relationship between requirements and risks be maintained. The RE tool should support risk management as follows:

- Exchanging information pertinent to risk management with external risk management tools
- Maintaining relationship of requirements to risks raised and risks mitigated

5.6.6 Project management

Project management applies to managing resources, schedules, and commitment of the requirements. While it is not the role of the RE tool set to perform project management, project management needs to keep track of the status of requirements. The RE tool should support project management as follows:

- Recording, tracking, and reporting the status of the overall requirements management process
- Exchanging information pertinent to project management with external project management tools

5.7 Other tool capabilities

5.7.1 Overview

The output of each RE activity and task is the basis for all subsequent systems and software life cycle activities including project planning, design, and coding, as well as the foundation for testing and user documentation. Therefore, integrating RE tools into systems and software development environment is important for automated RE support.

5.7.2 Collaborative access

In a distributed development environment, the synchronization of developers and access permissions to the requirement artifacts are key functions for ensuring integrity. The RE tool should support role-based collaborative access as follows:

- Maintaining data integrity under simultaneous multi-user access
- Assigning roles to users and groups of users
- Assigning role-based access to sets of requirements
- Assigning role-based access to attributes of requirements
- Assigning role-based access to tasks
- Maintaining the role-based access schema

5.7.3 RE tool administrative information

In many projects, versions can be distributed on different hardware platforms. In this case, administrative information will be needed to monitor system wide use. The RE tool should support this information as follows:

- Providing project Information (e.g., project size, concurrent users, and number of analyst)

- Providing user information indicating single user or multiple concurrent user environment
- Providing platform information indicating which platforms and operating systems support the tool
- Providing database information indicating which database it uses (proprietary or commercial)
- Providing hardware and/or software information indicating hardware and/or software configuration requirements (e.g., memory requirements, CPU requirements, disk space requirements, browsers)

5.7.4 Graphical user interface

To improve the usability of the tool, various capabilities of GUI should be supported. The RE tool should support graphical user interface as follows:

- Providing standard windows, Unix, Mac environment capabilities (e.g., multi-widows, open multiple files concurrently)
- Providing multiple windows or views into the tool (e.g., a change in one view will automatically reflect in all other views)
- Providing an interactive graphical input and manipulation of data
- Enabling web browser interface
- Providing edit, undo, cut, and paste function
- Providing an ability for doing one thing while users are looking at another (e.g., run a report and look at a requirements at the same time)

6 Quality characteristics of requirements

6.1 Overview

Setting up a target quality criteria (or quality characteristics) can help in the early detection and correction of problems. The quality characteristics can also be used as a metric in evaluating the requirements.

This section describes the sets of quality characteristics of requirements and how the RE tool can support in defining and utilizing them.

6.2 Quality characteristics of requirements artifacts

The associated quality characteristics for individual requirement statements are denoted as follows:

- Singularity – each requirement statement addresses a single concern at an agreed granularity
- Correctness – requirement statement corresponds to user request
- Conciseness – brief meanings, no unnecessary information
- Non-ambiguity – requirement statement has a single interpretation
- Traceability – requirement statement is upwards and downwards traceable to surrounding artifacts
- Compliancy – whether or not the requirement is mandatory
- Priority – the urgency of the requirement is determined

- Verifiability – requirement statement is testable
- Abstraction – requirement statement is expressed at the proper level of detail
- Feasibility – requirement statement is technically feasible
- Stability – requirement statement is not recently subject to frequent changes
- Sufficiency – requirement statement is sufficiently covered in each dimension of downwards traceability (e.g. satisfaction traceability down to lower requirements, and verification traceability across to tests)
- Necessity – requirement statement is necessary in each dimension of upwards traceability in which it engages (e.g. the requirement is necessary to satisfy higher requirements)

The associated quality characteristics for sets of requirement statements are denoted as follows:

- Consistency – absence of conflict among a set of requirements
- Uniformity – consistent use of terms across a set of requirements
- Modularity –cohesion among a set of requirements (i.e. requirements that belong together are together)
- Scoped – requirement set within the defined scope
- Uniqueness – all requirement statements in set are significantly different (i.e. no similar or identical requirements)
- Completeness – the set of requirements represents a complete definition of the product
- Understandability - the set of requirements is comprehensible as a document
- Affordability – the set of requirements is attainable with respect to schedule and cost

6.3 Tool capabilities for quality characteristics

6.3.1 Overview

The RE tool should support the effective use of quality characteristics in classifying and assessing the requirements as follows:

- Supporting stakeholders in identifying, specifying and quantifying key quality characteristics of requirements
- Providing services to specify, measure and review requirements using the identified quality characteristics
- Allowing the user to ensure that the requirements have the necessary quality characteristics needed by the stakeholders
- Allowing the user to record and report the status of each requirement with respect to each quality characteristic
- Allowing the user to record and report rationale associated with the assessment of a requirement statement with respect to each quality characteristic

In addition to the above capabilities, the attribute-specific capabilities given in 6.3.2 to 6.3.22 should be provided.

6.3.2 Singularity

A requirement statement should address a single aspect. A single requirement may consist of multiple elements, such as a statement of capability, one or associated modes of operation and one or more associated levels of performance. The RE tool should support detection of multiple elements as follows:

- Providing a numbering or naming scheme for uniquely identifying a requirement or element of a requirement
- Indicating the presence of multiple statements that combines two or more elements

NOTE Completeness and singularity need to be balanced, and modularity has to be used to ensure that related requirements are grouped.

6.3.3 Correctness

The correctness of requirement with respect to stakeholder needs is addressed by the requirements validation activity. The RE tool should support validation as follows:

- Supporting a validation process in which requirements can be reviewed with respect to stakeholder needs

NOTE The use of mathematical formalisms can address certain issues of correctness of specifications against requirements. There is no known method of automatic correctness checking of requirements against stakeholder needs.

6.3.4 Conciseness

To achieve conciseness, requirements should be expressed as single sentences with a small number of sub-clauses. The RE tool should support validation as follows:

- Identifying requirements that break the conciseness rule

6.3.5 Non-ambiguity

The RE tool should support the detection of ambiguity as follows:

- Providing a user-manageable list of inherently ambiguous terms, expressions, and syntactic structures
- Highlighting to the user occurrences of ambiguous terms, expressions and syntactic structures in requirements statements
- Allowing the user to record exceptions in the use of listed ambiguous terms, expressions and syntactic structures (e.g. state the rationale for allowing the use of such terms in particular circumstances.)

NOTE Some requirements specification languages would allow a tool to support automatic detection of lexical, syntactic, and semantic errors over and above those detectable by searching for words and phrases.

6.3.6 Traceability

The primary purpose of traceability is to enable impact analysis. It is not only requirements that engage in such traceability; it may be other development artifacts, such as test cases, issues, and risks. It should be possible to discover what artifacts may be impacted if a selected artifact were to change. The RE tool should support both backward and forward traceability. Backward traceability links each artifact explicitly to what

impacts it in previous stages of development; for instance a system requirement back to the stakeholder requirements it is intended to satisfy. Forward traceability links each artifact to what it impacts in subsequent stages of development; for instance, a system requirement to the tests that are designed to validate it. The RE tool should support backwards and forwards traceability as follows:

- Allowing the user to define valid traceability relationships (i.e. which artifacts in which documents may be linked)
- Allowing the user to characterise traceability relationships by name and cardinality
- Allowing the user to create valid backwards traceability from an artefact to those that impact it
- Allowing the user to create valid forwards traceability from an artefact to those that it impacts
- Allowing the user to report on multi-level backwards and forwards impacts from selected artifacts

6.3.7 Compliancy

Compliance refers to an attributes of a requirement which expresses the level or degree of compliance to a standard. There are two levels of compliance characteristics to standards:

Mandatory: The standards are applicable to all requirements defined in sentences containing the word “shall”. These requirements must be verified.

Goal or objective: The standards are not mandatory but need rationale and documentation if requirements are not met by the standards. The RE tool should support for:

- Allowing the user to define a number of compliance levels (e.g. mandatory or desirable)
- Allowing the user to classify each requirement with a compliance level

6.3.8 Priority

Establishing the relative priority of each requirement enables users to develop against the highest priority requirements first. The RE tool should support requirements prioritization as follows:

- Providing a capability for specifying the importance of individual requirements

6.3.9 Verifiability

A requirements is verifiable if there exists method by one can test that the product meets the requirements. If a method cannot be devised, the requirement should be removed or revised. The RE tool should support verifiability as follows:

- Providing a user-manageable list of inherently vague terms, expressions, and syntactic structures that render a requirement unverifiable
- Highlighting to the user occurrences of vague terms, expressions and syntactic structures in requirements statements
- Allowing the user to record exceptions in the use of listed vague terms, expressions and syntactic structures (e.g. state the rationale for allowing the use of such terms in particular circumstances)
- Allowing the user to associate a selection of verification methods with each requirement
- Allowing the user to associate a rationale against the set of test methods associated with each requirement

6.3.10 Abstraction

Each requirement should be positioned at an appropriate level within the system hierarchy. For example, level I may indicate a top or system level requirements. Level II may be the segment level requirements. The RE tool should support abstraction as follows:

- Identifying levels within the system hierarchy
- Allowing the positioning of requirements within appropriate levels
- Supporting a review process that checks the appropriate level of abstraction of a requirement

6.3.11 Feasibility

The RE tool should support feasibility as follows:

- Supporting a requirements review process that assesses the feasibility of each requirement

6.3.12 Stability

The earlier a requirements stabilizes, the less the test plans change. An unstable requirement is one that is changed frequently. The RE tool should support abstraction as follows:

- Identifying and monitoring the requirements with frequent changes and outdated test cases

6.3.13 Sufficiency

The RE tool should support sufficiency as follows:

- Allowing the user to capture a rationale for the sufficiency of the set of downwardly linked requirements that satisfy each requirement
- Supporting a review process that determines the sufficiency of the set of downwardly linked requirements that satisfy each requirement

6.3.14 Necessity

The RE tool should support necessity as follows:

- Allowing the user to capture a rationale for the necessity of each requirement for each upwardly linked requirement it serves to satisfy
- Supporting a review process that determines the necessity of each requirement for each upwardly linked requirement it serves to satisfy

6.3.15 Consistency

The challenge here is identifying conflict among a large number of requirements. Consistency is easier to check among small sets of related requirements. Thus, knowing which requirements are related in some way is useful. The RE toolset should support the detection of conflicting requirements as follows:

- Allowing the user to identify related requirements through detection of keywords
- Allowing the user to identify related requirements through classification of requirements in multiple ways
- Presenting sets of related requirements for conflict review