

**ISO/IEC JTC 1
Information Technology**

Document Type: Text for DTR Ballot

Document Title: DTR18018 Information Technology — Configuration Management Tool Requirements.

Document Source: JTC 1 Secretariat

Reference:

Document Status: This document is circulated to JTC 1 National Bodies for a 3 month DTR ballot. National Bodies are asked to vote and submit their comments via the on-line balloting system by the due date indicated.

Action ID: VOTE

Due Date: 2009-04-24

No. of Pages: 57

ISO/IEC JTC1/SC7 /N4187

2009-01-05

Document Type	DTR
Title	DTR18018 Information Technology — Configuration Management Tool Requirements.
Source	WG4
Project	18018
Status	DTR
Reference	N3886, N4011, Resolution 1040, Resolution 1042, N4187
Action ID	FYI
Distribution	AG
No. of Pages	36
Note	Sent to JTC 1 for DTR ballot.

Address reply to: ISO/IEC JTC1/SC7 Secretariat
École de technologie supérieure – Department of Software and IT Engineering
1100 Notre Dame Ouest, Montréal, Québec Canada H3C 1K3
secretariat@jtc1-sc7.org

www.jtc1-sc7.org

ISO/IEC JTC1/SC 7/WG4 N 439

2008-11-7

ISO/IEC DTR 18018

ISO/IEC JTC1/SC 7/WG 4

Secretariat: Standard Council of Canada

Information Technology – Guide for Configuration Management Tool Capabilities

Warning

This document is not an ISO International Standard. It is distributed for review and comment. It is subject to change without notice and may not be referred to as an International Standard.

Recipients of this draft are invited to submit, with their comments, notification of any relevant patent rights of which they are aware and to provide supporting documentation.

Document type: Technical Report
Document subtype:
Document stage: DTR
Document language: E

K:\ISO\4N_439 18018 Draft DTR.doc

Copyright notice

This ISO document is a working draft or committee draft and is copyright-protected by ISO. While the reproduction of working drafts or committee drafts in any form for use by participants in the ISO standards development process is permitted without prior permission from ISO, neither this document nor any extract from it may be reproduced, stored or transmitted in any form for any other purpose without prior written permission from ISO.

Requests for permission to reproduce this document for the purpose of selling it should be addressed as shown below or to ISO's member body in the country of the requester:

[Indicate:

the full address

telephone number

fax number

telex number

and electronic mail address

as appropriate, of the Copyright Manager of the ISO member body responsible for the secretariat of the TC or SC within the framework of which the working document has been prepared.]

Reproduction for sales purposes may be subject to royalty payments or a licensing agreement.

Violators may be prosecuted.

Contents

1	Scope	1
2	Normative references	1
3	Terms and definitions	1
4	Application of this technical report	3
5	Capabilities of configuration management tools	4
5.1	Overview of configuration management tool capabilities	4
5.2	Configuration management tool capabilities	4
5.2.1	Configuration identification	4
5.2.2	Configuration baselining	4
5.2.3	Configuration control	4
5.2.4	Configuration status accounting	5
5.2.5	Configuration auditing	5
5.2.6	Release management and delivery	5
5.2.7	Other configuration management tool capabilities	5
5.3	Configuration identification	5
5.3.1	Overview	5
5.3.2	Identifying configuration items	6
5.3.3	Specifying configuration item relationship	6
5.3.4	Defining states and specifying status of configuration items	6
5.4	Configuration baselining	7
5.4.1	Overview	7
5.4.2	Creating a baseline	7
5.4.3	Storing baseline	7
5.4.4	Tracing baseline	7
5.4.5	Reporting baseline status	7
5.5	Configuration control	8
5.5.1	Overview	8
5.5.2	Change control	8
5.5.3	Access control	9
5.5.4	Version control	10
5.6	Configuration status accounting	12
5.6.1	Overview	12
5.6.2	Recording status information	12
5.6.3	Tracing the status	12
5.6.4	Configuration status reporting	12
5.7	Configuration auditing	13
5.7.1	Overview	13
5.7.2	Audit planning and procedures	13
5.7.3	Executing auditing	13
5.7.4	Reporting auditing results	14
5.7.5	Analyzing anomaly	14
5.8	Release management and delivery	14
5.8.1	Overview	14
5.8.2	Building	14
5.8.3	Storing	15
5.8.4	Replicating	15
5.8.5	Packaging and delivery	15
5.8.6	Tracking the distribution of products	15
5.9	Other configuration management tool capabilities	16
5.9.1	Integrating with other tools and platforms	16

Annex A (informative) **Focus areas of each reference** 17

Annex B (informative) **Configuration management services**..... 20

Bibliography..... 27

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 18018, 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

Configuration Management (CM) is a process central to the software engineering life cycle. CM has been established as an ISO/IEC standard life cycle process in ISO/IEC 12207:1995 - Information Technology – Software Life Cycle processes and ISO/IEC 15288: 2002 - Information Technology - System Life Cycle processes.

IS 12207 and IS 15288 describe a comprehensive set of processes, activities and tasks to be performed when acquiring or developing software. However, these documents do not address the capabilities that a CM tool user can expect from a tool in order to support the CM process and other software engineering life cycle activities. There is a gap between CM process descriptions and corresponding CM process automation services which affects both tool users and tool suppliers.

The outcomes of this technical report are to:

- 1) Provide guidance in the evaluation and selection for CM tools during acquisition. CM tool evaluation by prospective users can be complex, time consuming, and expensive. This standard helps to characterize what a CM tool can do or cannot do in the CM process.

- 2) Provide guidance for tool manufactures in implementing a minimum set of capabilities. The capabilities defined in this standard are linked to IS 12207 and IS 15288, and will provide tool manufacturers guidance as to what characteristics their tools should support to meet these ISO/IEC standards.

Information Technology – Guide for Configuration Management Tool Capabilities

1 Scope

Configuration management has been established as an ISO/IEC standard life cycle process in both ISO/IEC 15288: 2002 – Systems Engineering – Systems life cycle processes and ISO/IEC 12207:1995 - Information Technology – Software life cycle processes.

This technical report provides guidance for the configuration management tool capabilities from which systems and software development life cycle activities can be supported.

Although a method for evaluating general CASE tools has been adopted as an ISO/IEC standard, ISO/IEC 14102:1995 - Information technology - Guideline for the evaluation and selection of CASE tools, the standard only describes a set of evaluation criteria for tools without referencing a specific activity or task which the tool supports. This lack of consideration on a specific activity or task causes users confusion and difficulty in evaluating and selecting the right tools.

It is the scope of this technical report to supplement ISO/IEC 14102:1995 by providing a set of minimum tool capabilities for configuration management. It can be used as the set of criteria by a potential user during an acquisition process, or by a configuration management tool supplier to help identify desirable tool capabilities.

2 Normative references

ISO/IEC 12207:2008 Information technology – Software life cycle processes

ISO/IEC 15288:2008 Systems engineering – System life cycle processes

3 Terms and definitions

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

3.1

Attributes

A property associated with a set of real or abstract things that is some characteristic of interest..

3.2

Baseline

A version of a configuration, specification, or product that has been formally reviewed and agreed upon, that thereafter serves as the basis for further development, and that can be changed only through formal change control procedures.

[ISO/IEC 12207:2008]

3.3

Branch

Deviations from the main development line for a configuration item. This allows different person working on the same item at the same time.

3.4

Build

The process of generating (archiving) an executable and testable system from source versions or baselines. The build needs to compile and link the various versions, in the correct order. The build tools may be integrated into a configuration management tool.

3.5

Change request

CR

A formal procedure for submitting a request for an adjustment of configuration item.

3.6

Configuration item

An entity within a configuration that satisfies an end use function and that can be uniquely identified at a given reference point.

[ISO/IEC 12207:2008]

3.7

Configuration management

CM

Coordinated activities to direct and control configuration.

3.8

CM services

Abstract description of work done by CM tools. A service is self contained, coherent, discrete, and may be composed of other services.

3.9

CM tool

A software product that can assist software engineers by providing automated support for configuration management activities.

3.10

Configuration status accounting

CSA

An element of configuration management, consisting of the recording and reporting of information needed to manage a configuration effectively.

[ISO/IEC 24765]

3.11

Delta

The difference between two versions.

3.12

Software/System elements

Element that define and prescribe what a software or system is composed of (for example, requirements, design, code, test cases, and version number). An element can contain sub elements, or other software/system elements that are dependent on the top level element.

3.13

Release

A particular version of a configuration item that is made available for a specific purpose (for example, test release).

3.14**Traceability**

The degree to which each element in a software development product establishes its reason for existing.

3.15**Version**

An identified instance of a configuration item.

Note Modification to a version of a software product, resulting in a new version, requires configuration management action.

3.16**Version identifier**

Supplementary information used to distinguish a version of a configuration item from other versions. Version numbers are used to compare the version of the software product against another version.

4 Application of this (draft) technical report

4.1 Overview

This clause presents the benefits to groups of people that acquire, supply, develop, operate, and maintain a CM tool. The objective is to provide a road map for the users of this (draft) technical report so that they can orient themselves in it and apply it judiciously.

4.2 CM personnel

Personnel involved in the performance of one or more CM activities will benefit from this technical report as follows:

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

4.3 Tool suppliers

Suppliers of software engineering tools will benefit from this technical report as follows:

- develop CM tools consistent with the International Standards IS 12207, IS 15288, and IS 14102
- provide CM tools that can be shown to support an internationally accepted set of capabilities

4.4 Acquirers

People involved in the purchase of CM tools will benefit from this technical report as follows:

- review CM services that can contribute to CM process improvement
- identify criteria for selecting CM tools

- compare competing CM tools based upon this technical report

5 Capabilities of configuration management tools

5.1 Overview of configuration management tool capabilities

Throughout the systems and software life cycle, different artifacts (e.g., hardware, software, documents) exist in different versions at different times and changes arise constantly. Configuration Management (CM) verifies and assures that a product performs as intended by providing visibility and control of product's functional and physical characteristics. A CM tool can provide automated assistance for CM activities: configuration identification, change management, reports, status accounting, and auditing. This technical report provides the tool capabilities for the automation of CM activities to support software and systems lifecycle processes.

NOTE The following documents have been reviewed for identifying the CM tool capabilities: IS 12207, IS 15288, TR 15846, TR 19759 SWEBOK, IS 14102, IS 10007:2003, IS 15940:2006, IEEE 828:1990, ANSI/EIA 649:1998, and commercial tool information. The life cycle processes and activities from these references have been used as the basis for the CM tool capability categorization found in this document.

5.2 Configuration management tool capabilities

The following is the CM activities described in 12207 and 15288; The CM tool should provide the capabilities supporting these CM activities:

5.2.1 Configuration identification

Configuration identification should capture the attributes of the software to be controlled: the software contents, the different versions of contents, the operation data, and any other essential elements that constitute the configuration item. The tools should provide the following capabilities for the configuration items:

- identifying configuration items
- specifying configuration item relationship
- defining states and specifying status of configuration items

5.2.2 Configuration baselining

For each configuration item and its versions, the tool should provide the following capabilities for baselines:

- creating baseline
- storing the baseline with its version references and identification details
- tracing baselines
- reporting baseline status

5.2.3 Configuration control

The tool should provide the following capabilities for configuration control:

- controlling changes
- controlling access

- controlling versions

5.2.4 Configuration status accounting

The tool should provide the following capabilities for status accounting:

- recording status information
- tracing the status
- reporting status of configuration management items

5.2.5 Configuration auditing

The tool should provide the following capabilities for configuration audits:

- planning audit
- auditing the functional completeness of configuration items against requirements
- auditing the physical completeness of the configuration items (whether their design and code reflect an up-to-date technical description)
- reporting audit results
- analyzing anomaly

5.2.6 Release management and delivery

The tool should provide the following capabilities for release management and delivery:

- building software products
- storing the master copies of the code and documentation over the life of the software product
- replicating, packaging, and delivering of code and documentation in accordance with the policies of the organizations involved
- tracking build and release identification

5.2.7 Other configuration management tool capabilities

The tool should provide the following general capabilities:

- integrating with other tools and platforms

5.3 Configuration identification

5.3.1 Overview

Configuration identification is the basis for the subsequent control of the software configuration. This activity includes identifying items to be controlled, establishing an identification scheme for the items and their versions, and specifying the relationship and status of the configuration items.

5.3.2 Identifying configuration items

Configuration item identification should capture at a minimum the following characteristics of the software/system elements to be controlled:

- software/system element contents (e.g., requirements, design, analysis, code, test cases, and traceability between the elements)
- different versions of configuration item and its constituent components
- software/system element operation data (e.g., history of change, version number, number of change requests)
- name of configuration item

The tool should provide the following capabilities for configuration item identification:

- generating a unique identifier to distinguish each item (e.g., alphabetic, alphanumeric)
- providing a means of documentation (e.g., templates or forms) of the configuration information to be used to describe each configuration item (e.g., configuration identification number, link to other configuration item, link to software structure, link to baseline, link to version hierarchy, link to storage, owner, creation date, version number, and configuration status)
- linking to change requests associated with the configuration item

5.3.3 Specifying configuration item relationship

The structural relationships among the configuration items affect other CM activities or tasks, such as software building or analyzing the impact of proposed changes. The tool should provide the following capabilities for configuration item relationships:

- the tracing to and from the configuration items
- the tracing the configuration item relationship links both upward and downward in the software hierarchy
- the mapping of the identified configuration items to the elements of software structure (e.g., system, sub-system, component, library, or unit code)

5.3.4 Defining states and specifying status of configuration items

A configuration item may have a lifecycle that goes through different states (e.g., checked-in, checked-out, change-requested, change-approved, change-rejected, change-reviewed, change-tested, and change-completed). Changes in state are triggered by conditions or events (e.g., submit, retrieve, change-request-submit, change-approve, change-review, change-reject, testing, and change-complete).

The status of a configuration item is defined by the state in which it exists at any given time.

The tool should provide the following capabilities for configuration item status:

- defining the set of states of configuration items
- defining the conditions or events that enable or enact the change of configuration item status
- specifying the status of configuration items

5.4 Configuration baselining

5.4.1 Overview

A baseline is a set of configuration items designated at a specific time during the software life cycle. It represents the approved status of configuration items and provides the point of departure for further evolution. A configuration item belongs to more than one baseline. Baselining activities include the creating, storing, tracing, access controlling, and reporting of baseline.

5.4.2 Creating a baseline

Creating a baseline consists of identifying which configuration items are part of it. The tool should provide the following capabilities for initiating baselines:

- naming the configuration items and versions specified in the configuration item hierarchy that comprise the baseline

5.4.3 Storing baseline

The baseline must be stored with the detailed attributes associated with acceptance, change, support, and disposal of baseline. The tool should provide the following capabilities for storing baselines:

- placing the baseline attributes into a repository, (e.g., creation date, owner, acceptance criteria, and change history)
- the placing of associated tool information for versioning, build, and release
- searching the attribute information of baseline from the repository
- changing the attribute information from the repository

5.4.4 Tracing baseline

Traceability is a key capability that helps investigate the history of every change in a baseline chain. The tool should provide the following capabilities for tracing baselines:

- the tracing across baselines (e.g., the system requirements baseline, software requirements baseline, design baseline, and delivery baseline)
- tracing the systems and software artifacts including derived ones (e.g., the project plan, requirement definition document, standards, system analysis document, system design document, prototype, system test plan and specification, program source code, object code and executable, unit test plan and specification, test and project data, and user manuals)

5.4.5 Reporting baseline status

The status report on baseline should include a range of baseline information. The tool should provide the following capabilities for reporting baseline status:

- searching for the baseline information at a minimum by configuration item, change history, status, release identifier, and links to other baselines
- generating a status report on a baseline at a minimum by configuration item, change history, status, release identifier, and links to other baselines
- reporting change requests associated with a baseline

5.5 Configuration control

5.5.1 Overview

Software configuration control is concerned with managing changes during the software life cycle. It covers the activities for controlling changes, controlling access to the configuration item, and controlling versions resulting from the modification of configuration items.

5.5.2 Change control

Change control covers the activities for determining what changes to make, approving changes, and implementing changes as follows:

- proposing the change request
- analyzing the changes
- approving or disapproving the request
- implementing and releasing the changes
- communicating the disposition
- tracking the changes

5.5.2.1 Proposing changes

Change request needs formal procedures for submitting and recording the anomaly or enhancement from the baselined configuration item. The tool should provide the following capabilities for proposing changes:

- providing a template or a form for change request which includes at a minimum, configuration and version identification number, requester, date of request, reason for change, priority of the proposed changes
- storing the change request information into a repository and notifying the request to the authorized person by using email, groupware, or any other electronic communication means
- querying and tracing the change request at a minimum, by requester, date, configuration identification number, and reason for change

5.5.2.2 Analyzing changes

Impact analysis is required to determine the extent of modifications that the change request. The tool should provide the following capabilities for analyzing changes:

- tracing configuration items and related baselines affected by the proposed change
- tracing any approved modifications affecting the identified configuration items or baselines

5.5.2.3 Approving/disapproving change requests

The authority for accepting or rejecting proposed changes depends on a Configuration Control Board (CCB). The board reviews each proposed change, approve or disapprove it, and if approved, coordinate the change with the affected group. The tool should provide the following capabilities to support the CCB:

- providing a template or a form for describing the scope and reason of the change (e.g., accept, modify, reject, or defer)

- enabling to specify the associated information (e.g., change priority, change effort, approver, responsible persons for the change, due date for the change implementation)
- providing a means of recording the decision of the CCB

5.5.2.4 Implementing/releasing the changes

After the change request is approved, modification should be performed on only the approved modification. To implement and release each approved change, the tool should provide the following capabilities:

- tracking of which change requests are incorporated into particular software versions and baselines
- allowing only the versions and baselines which are associated to the approved modifications be changed in a workspace
- monitoring that the changes are completed on time and are implemented by the authorized person
- releasing the changed versions to the designated medium and destination
- the roll back of a configuration item so that at any point, a configuration item can be brought back to its previous state

5.5.2.5 Communicating disposition of change requests

The designated changes should be distributed by notifying everyone impacted by the disposition. The tool should provide the following capabilities for communication dispositions:

- notifying those who use the affected configuration items if the change is approved
- notifying those who proposed change if the change is rejected or deferred

5.5.2.6 Tracking changes

Managing the audit trail for each configuration item is important in controlling the changes. The tool should provide the following capabilities for tracking changes:

- tracking the configuration items and versions by change attributes (e.g., the reasons of change, change request date, change requester, change completion date, configuration identification number, or change status)
- tracking the status of the modification (e.g., implemented, verified, released)
- representing the status of changes in a reporting form (e.g., table, graph, or chart)

5.5.3 Access control

Access control ensures the protection of information and data so that unauthorized persons or systems cannot read or modify the information and, at the same time, authorized persons or systems are not denied access to the information and data.

5.5.3.1 Specifying and maintaining access

Controlling access to configuration items starts with specifying the access permission and security levels. The tool should provide the following capabilities for specifying and maintaining access:

- specifying different access permissions to configuration items with different status (e.g., login, check in, check out for read, check out for modification, change request, change approved/rejected, change implemented, release, audit)
- specifying different access permissions by user, group, or role
- specifying different access permissions to different granularities (e.g., line, code, function, module)
- maintaining the log for all the access to the configuration item by permission and granularity
- providing an encryption method to ensure secure usage and transmission of data

5.5.3.2 Concurrency control

Different users should be able to gain access concurrently to the configuration information. The tool should provide the following capabilities for concurrent control:

- producing the latest version of a branch when a version is retrieved for modification
- ensuring that only one person at a time can create a new version using a locking mechanism, forcing serialized changes to any given version
- enabling parallel concurrent development by allowing multiple users access to different versions at the same time

5.5.4 Version control

A version consists of a file or a set of files, each with a particular version identification number. Version control maintains multiple versions or a set of related versions of an application. The fundamental operations associated with version control are version creation, version modification, version merge, and version comparison.

5.5.4.1 Version creation

A new version should be created if changes are anticipated which should not affect the existing version.

The tool should provide the following capabilities for version creation:

- identifying the version of the configuration item to be changed
- placing a new version into the version branches (e.g., sequential, multiple)

5.5.4.2 Version modification

Version modification allows an existing version to evolve by creating a new version.

The tool should provide the following capabilities:

- a means of modifying a version by creating a new version.
- a means of preventing simultaneous conflicting modifications by different users to the same version of a configuration item.

NOTE: The challenge here is in a multi-user environment in which several users may wish to make simultaneous changes to the same version. A common solution to this challenge is to use a check-in/check-out mechanism, which prevents a configuration item from being edited by multiple users. Other mechanisms allow multiple changes that are later merged.

5.5.4.3 Merging versions

Merging is the act of reconciling multiple changes made to different version of the same file. The changes are merged, resulting in a single new version that contains both sets of changes. In some cases, the merge can be performed automatically, e.g., the changes don't conflict. In other cases, a person must decide exactly what the resulting version should contain. The tool should provide the following capabilities for merging versions:

- merging the content of designated versions
- propagating the merge result to the subsequent versions along the version tree
- recording the merge history in the repository (e.g., date and time, owner, types of merge, location of changes, new version identification number)
- placing a new merged version in the version tree or graph

5.5.4.4 Comparing versions

Two versions of a software/system element may be statically compared, providing a list of the differences, called deltas.

The tool should provide the following capabilities for managing deltas:

- comparing two versions of a software/system element and identifying the deltas
- undoing selected deltas
- applying selected deltas to other versions of the same software/system element

5.5.4.5 Change history

A record may be kept of the sequence of changes made to a version of a software/system element, This information may then be used to roll back a sequence of changes within the version.

The tool should provide the following capabilities for managing change history:

- recording the sequence of changes made to a software/system element
- reporting the sequence of changes made to a software/system element
- undoing changes

5.5.4.6 Granularity

Configuration items will vary in size and granularity (e.g., line, functions modules or files).. The appropriate granularity should be specified and document for version control. The tool should provide the following capabilities for configuration item granularity:

- specifying the granularity of version, access permission, delta, and merge (e.g., line, functions, modules, files)
- specifying the granularity changes on version, access permission, delta, and merge

5.6 Configuration status accounting

5.6.1 Overview

Configuration status accounting ensures that configuration status is recorded, monitored, and reported. Configuration status can be identified and managed by tracing the changes to a configuration item in the chain.

5.6.2 Recording status information

To enable the configuration status accounting, the configuration item status information should be identified and maintained. The tool should provide the following capabilities for recording status information:

- recording the identification and status of each new and modified configuration item and of baseline
- recording the version and status of each modification (e.g., version/release identifiers, change requested, and change approved)

5.6.3 Tracing the status

To capture and search the status and history of configuration items, the tool should provide the following capabilities:

- tracing the configuration and version identification number (e.g., number of changes to the latest versions, comparisons of versions, the number of releases)
- tracing the other documents affected by each change and update
- tracing the implementation of approved modifications

5.6.4 Configuration status reporting

Reported information can be used by various organizations and project elements. Reporting can take the form of ad hoc queries or the periodic production of pre-designed reports. The tool should provide the following capabilities for reporting and status accounting:

- change/problem tracking and reporting (e.g., who made the change, when initiated the change, change history, and change request (CR) status)
- allowing users to tailor the information in various formats (e.g., all unassigned CRs, all pending CRs, all CRs assigned to a particular person, CRs sorted by date, severity, priority, classification, completion date, status, and other criteria)
- generating difference reports containing the differences between two versions of an item
- generating release reports containing the release identification and status
- providing event or transaction log with the history of CM activities performed (e.g., transaction number date, originator, nature of the entry, affected items, activity, description, participants, impacted items, and remarks)
- allowing users to write queries to get the configuration information in their own format
- tracking among change requests to source problems (e.g. a failed test may generate multiple change requests)

5.7 Configuration auditing

5.7.1 Overview

Configuration auditing verifies that the software system matches the configuration item specification and that the product package contains all of the components it is supposed to contain and performs as promised. Effective auditing should support the verification and validation processes to ensure completeness and integrity of software products (functional, physical, baseline auditing). Configuration auditing consists of activities including audit planning, executing auditing, reporting auditing results, and resolving anomalies.

5.7.2 Audit planning and procedures

Audit plan describes all CM aspects (e.g., resources and procedures) that the auditing must include. The tool should provide the following capabilities for the audit planning:

- specifying unique identifiers of configuration items and their current status
- specifying configuration baselines, releases, and their status
- identifying the latest configuration item versions and reporting their status
- specifying the person responsible for each CM activity (e.g., configuration identification, baselining, change control, status accounting, and auditing)
- specifying build instruction and tools used to develop, produce, test, and verify the product
- tracing change history and generating an audit trail

5.7.3 Executing auditing

Auditing activity determines the extent to which an item satisfies the required functional and physical characteristics. Audits check the completeness of the software, system, or product.

5.7.3.1 Functional configuration auditing

Functional auditing verifies that a configuration item's actual performance agrees with the requirements specified in the requirements definition and system design documents. The tool should provide the following capabilities for functional audits:

- tracing and maintaining input documents for testing that include test plans and test data
- test methods to trace that all functional parameters that were tested and test results

5.7.3.2 Physical configuration auditing

Physical auditing is to verify that a configuration item conforms to the technical documentation that defines it. The tool should provide the following capabilities for physical audits:

- tracing across configuration items, versions, links to software structure, and release structure
- specifying the product baseline
- checking that the software product specification and version identification are consistent with the real software product (e.g., source code, user documents, or any other configuration items)

5.7.4 Reporting auditing results

After completing an audit, the audit results should be documented and reported. The audit results contain any problems found in the audit and related problem resolutions planned. The tool should provide the following capabilities for reporting audit results:

- searching and tracing the configuration items, build, or release by configuration information (e.g., identifiers, date, change requester)
- searching and tracing the latest configuration item versions for a system build and application
- reporting the metric statistics (e.g., number of (approved) change requests for a system, the number of baselines and releases, the usage of configuration items in build and release, average time taken for the resolution of change request, time spent on development, testing, number of defects found after each release, time taken to identify the source of a defect, time taken to resolve a defect, amount of reused code, and the number of configuration items impacted by a change request)
- comparing baselines and releases, and reporting the differences
- report anomalies
- customizing report generation and exporting the reports to other applications such as word processors, spread sheets

5.7.5 Analyzing anomaly

After the execution of audit, any anomaly reported need to be examined and analyzed for a future resolution. The tool should provide the following capabilities for anomaly/problem resolution:

- tracking the anomaly/problem from its origin and capture the details of the problem, (e.g., as originator, date of problem identification, cause of the problem, how and when it was fixed, how much time was taken, and what kind of skill were needed)
- reporting the identified anomaly to the person responsible for coordinating the anomaly/problem resolution
- creating automatic alerts based on the discovery of the anomaly/problem
- tracing all the change requests that are associated with particular configuration item , reasons for change, change request status, change result, and the configuration items that the change affects
- providing a checklist or a form to ensure that all discovered anomalies are recorded, analyzed, and resolved
- storing problem resolution results into a repository so that users can look up the knowledge base to determine if the same problem has occurred

5.8 Release management and delivery

5.8.1 Overview

Release management refers to the distribution of a software configuration item outside the development activity. The release and delivery of software products and associated documentation requires that master copies of source code and associated documentation be maintained for the life of the software product.

5.8.2 Building

Capturing the details of every build for a system to be tested or released is an important activity. The tool should provide the following capabilities for building configuration items:

- tracing configuration items to find out which configuration items are used in the build and what versions of the configuration items are used
- grouping configuration items by marking the appropriate version of each configuration item with a symbolic name or identification number that is specific to the release
- scanning automatic source code
- maintaining the traceability of software artifacts so that the software products and development environment can be changed and reproduced in the future
- retaining the baseline for the software library and environment
- programming and running build scripts
- creating build audit trails

5.8.3 Storing

Before versions of the software products are released and delivered, the released versions and their associated documentation should be stored and maintained in a repository. The tool should provide the following capabilities for storing:

- maintaining the history of the previous builds and releases in the repository
- selecting storage medium
- marking the identification of the release in terms of physical medium (e.g., CD-ROM or magnetic tape) and electronic releases (e.g., download into operational libraries)

5.8.4 Replicating

Replication is a manufacturing stage for software by making copies. The tool should provide the following capabilities for replication:

- copying the configuration items, versions, and other related documents into a designated place
- checking that the release contains no extraneous items (e.g., software viruses or test data)
- preserving the integrity of the contents by freezing or locking the version during replication to prevent modification during or after delivery

5.8.5 Packaging and delivery

Packaging and delivery should be done by approved procedures and clearly marked with the release identification. The tool should provide the following capabilities for packaging and delivery:

- archiving a copy with the configuration items including the product information (e.g. license agreement and copyright statement).

5.8.6 Tracking the distribution of products

After the delivery of the product, the bug reports and change requests can be submitted from customers and systems. The tool should provide the following capabilities for tracking the distribution of the product:

- tracking the distribution of products to customers

- connecting the release management process to the change request process in order to map release contents to the change requests

5.9 Other configuration management tool capabilities

5.9.1 Integrating with other tools and platforms

Integrating CM tools into the software development environment is a key to providing automated CM support in software development environments. The tool should allow the integration of the CM tool with the software development environment by:

- providing integration with CASE tools, code generators, test data generators, automatic testing tools, and code analyzers
- supporting the interface with a variety of control and data integration
- separating the interface from the service mechanisms
- enabling the use of multiple implementations of environment services
- sharing the workspace with CASE tools so that users can perform relevant CM functions within the shared workspace

Annex A

(informative)

Focus areas of each reference

A.1 Configuration management focus area

Each of the following references describes some sets of CM capability:

ISO/IEC 12207:1995/Amd.1 12207:2002 Information technology – Software life cycle processes

- management of CM process
- configuration control
- configuration status accounting
- configuration evaluation (auditing)
- release management

ISO/IEC 15288:2002 Information technology - System life cycle processes

- configuration management strategy
- identifying items that are subject to configuration control.
- maintaining information on configurations with an appropriate level of integrity and security
- ensuring that changes to configuration baselines are properly identified, recorded, evaluated, approved, incorporated, and verified.

ISO/IEC 14102:2008- Information technology- Guideline for the evaluation and selection of CASE tools International Standard

- access control
- tracking of modifications
- definitions and management of multiple versions
- configuration status accounting
- release generation
- archival capability

ISO/IEC TR 19759 Software engineering – Guide to the software engineering body of knowledge (SWEBOK)

- management of CM process
- configuration identification
- configuration control
- configuration status accounting
- configuration auditing
- release management and delivery

ANSI/EIA 649 National Consensus Standard for Configuration Management

- configuration management planning and management
 - identifying context and environment

- configuration management plan
- implementation procedures
- training
- performance measurement
- supplier configuration management
- configuration identification
 - product information
 - product structure
 - product identifiers
 - documents identification
 - baselines
 - product identification recovery
 - interface control
- configuration control management
 - change identification
 - change evaluation and coordination
 - change implementation and verification
 - change management process applied to variances
- configuration status accounting
 - CSA information
 - CSA system
- configuration verification and audit
 - design and document verification
 - configuration audit
 - continuing performance audits and surveillance
- configuration management of digital data
 - digital data identification
 - data status level management
 - maintenance of data and product configuration relationships
 - data version control and management of review, comment, annotation and disposition
 - digital data transmittal
 - data access control

The following table summarizes the CM focal area from each reference:

Table 1 – CM capabilities in other references

CM capabilities	Focus area of each reference
Configuration process management (instantiating and defining , scope, plan, controlling execution, reviewing and evaluation. of processes, closing)	TR 19759 (1), ANSI/EIA 649 (1), 15288 (1)
Configuration identification	14102 (3), 12207 (1), 15288 (2), TR 19759 (2), ANSI/EIA 649 (2)
Configuration baselining	14102 (3), 12207 (1), 15288 (3), TR 19759 (2), ANSI/EIA 649 (2)
Configuration control (version control, access control, change control, release control)	14102 (2,5), 12207 (2,5), 15288 (3), TR 19759 (3,6), ANSI/EIA 649 (3)
Configuration reporting	14102 (4), 12207 (3), 15288 (4), TR 19759 (4), ANSI/EIA 649 (4)

Configuration auditing	14102 (4), 12207 (4), 15288 (4), TR 19759 (5), ANSI/EIA 649 (5)
Release management and delivery	14102 (5), 12207 (5), 15288 (4), TR 19759 (6), ANSI/EIA 649 (6)
Other requirements (e.g., integration with other tools, non-functional characteristics)	ISO/IEC 9126-1:2001 Software engineering – Product Quality – Part 1: Quality model IEEE 828:1990 IEEE Standard for Software Configuration Management Plans

Annex B (informative)

Configuration management services

B.1 Overview of service category

B.1.1 Introduction to CM as a service

Users may need to understand what services CM can provide for their software lifecycle activities. Better understanding and implementation of CM supports users in attaining a higher quality of product, more time for being productive on creative tasks, and better forecasting of software costs. Expressing the capabilities of CM via services emphasizes the fact that an individual performs some software lifecycle activities with the help of the CM. CM service provides a checklist of functionality for evaluating and choosing CM capabilities and for matching their process to a specific CM service. The service also can provide a specification of which CM mechanisms need to be in place in order to implement the certain functionality of lifecycle activity.

B.1.2 Rationale of CM services categorization

The services described in this annex reflect broad functional activities within a typical software engineering organization found in IS 12207, IS 15288, TR 15846, DTR 19759 SWEBOK, IS 14102, IS 15940, and IEEE 828:1990 IEEE Standard for Software Configuration Management Plans (Check appendix for the focus areas of each reference). The lifecycle processes and activities commonly found from these references have been analyzed and re-grouped from the view point of CM services as following:

- project management support services
- process improvement support services
- quality assurance support services
- reuse support services

B.1.3 Structure of service description

Each CM service is defined under two headings:

- Service concept, to provide a description of the service in terms that are not related to a specific implementation.
- Basic service operations, to list those operations that should be included in a service. This list of operational capabilities represents, in most cases, only primary services and is not intended to be complete.
- Automated service operations, provide a list of those operations that must be automated to be fully compliant to this technical report. This list is a subset of those operations listed under 'Basic service operations'. Partial compliance can be claimed if the tool implements one or more operations of this list.

B.2 Project management support services

The following services represent a collection of capabilities that provide support to a software project management process. Most of these services, except the risk management service, focus on the basic project management controls.

- requirement management service
- software project planning service
- software subcontract management service
- risk management service

B.2.1 Requirements management service

Service concepts

Requirements management is the process of establishing the common understanding on the project between the customer and the developer. Since the customer's requirement frequently changes throughout the software life cycle, maintaining, controlling, and tracing of the changes are important for the success of a project.

Basic requirements management service operations are as follows:

- identifying requirements baseline as configuration item
- modifying requirements baseline
- browsing requirements baseline
- reporting and displaying requirements baseline
- managing traceability between the requirement baseline and other documents
- auditing validation and verification of requirements

Automated requirements management service operations are as follows:

- baselined (customer) requirements are maintained
- changes to (customer) requirements are traced and change status is reported
- baselined requirements specifications are maintained
- changes to requirements specifications are traced, and change status is reported
- the dependencies between requirement documents and other documents is stored and traced

B.2.2 Software project planning service

Service concepts

Project planning service is the collection of discrete processes and activities needed to complete each of the processes by considering the resource availability and potential problems. Since the plan frequently changes throughout the software life cycle, maintaining, tracing, and reporting of the plan are important for the success of a project.

Basic project planning service operations are as follows:

- identifying resources, efforts, schedule, and costs estimation as baseline

- maintaining the baselined project plan
- managing changes to project plan
- identifying the role and work responsibility as baseline

Automated project planning service operations are as follows:

- historical estimation data and templates are provided
- Baselined project plans are maintained.
- changes to project plans are managed and change status is reported

B.2.3 Software subcontract management service

Service concepts

The focus of this service is to ensure that all contractual terms of the project is identified, documented, and maintained. Since a project governed by a contractual agreement is typically not complete until agreed to/by both parties, it is important that all contractual obligations have been described and controlled.

Basic subcontract management service operations are as follows:

- identifying the baseline of the subcontract statement of work
- identifying the baseline of the subcontract plan
- managing changes to the subcontract plan
- managing and browsing the output from the both parties
- performing the status reporting and auditing to maintain the communication channels for both parties

Automated subcontract management service operations are as follows:

- baselined subcontract statement of work is maintained
- changes to the subcontract statement of work are traced, and change status is reported
- the dependencies between the subcontract statement of work and other documents are stored and traced
- historical estimation data and templates are provided
- baselined subcontract plans are maintained
- changes to subcontract plans are managed and change status is reported

B.2.4 Risk management service

Service concepts

In addition to tracking the effort and schedule, the project manager makes plans to avoid or minimize any risk encountered during a software project. Since risk management entails planning and assessing the elements that are related to the success or failure of the project, maintaining and measuring all the resources that are relevant to the project is important.

Basic risk management service operations are as follows:

- identifying parameters for resource allocation and scheduling data as a baseline
- generating status reports for various allocation strategies
- maintaining traceability to support risk resolution

Automated risk management service operations are as follows:

- risk management plan is baselined
- historical data on risk management and resolution is provided
- changes to risk issues are managed and change status is reported

B.3 Process improvement support services

This section represents a collection of services that provide support to process improvements. These services are represented through a series of capabilities that provide mechanisms for institutionalization and implementation. Services for institutionalization include activities for establishing a process, process assessments, and continuous process improvement:

- process establishment service
- process assessment service
- continuous process improvement service

B.3.1 Process establishment service**Services concept**

This service provides capabilities for defining organizational processes. The service includes the composing and maintaining facilities for these processes. (e.g., analysis of current process, setting process improvement goals, defining process, developing action plan, and implementing processes)

Basic process establishment service operations are as follows:

- maintaining baseline for process definition
- managing changes to process definition
- supporting process audit
- maintaining process requirements and analysis data
- identifying, modifying, and reporting of process assets

Automated process establishment support service operations are as follows:

- process definition is baselined
- changes to process definition is managed
- process audit data is provided

B.3.2 Process assessment service

Services concept

This service provides capabilities to assess the state of an organization's current software process, to determine the high-priority software process-related issues facing an organization, and to obtain the organizational support for software process improvement.

Basic process assessment service operations are as follows:

- identifying and managing baseline of the assessment plan and procedures
- recording and reporting the assessment data
- managing changes to the assessment plan and procedures
- maintaining and reporting the occurrence of specific process events

Automated process assessment support service operations are as follows:

- historical estimation data and templates are provided
- baselined assessment plans are maintained
- changes to assessment plans are managed and change status is reported

B.3.3 Continuous process improvement service

Services concept

The lessons learned through the process establishment and the assessment service are used to evaluate and improve the existing process. Since the process improvement service is coordinated across the assessment, measurement, and modifications of the processes and project life cycles, maintaining and providing all the data that are relevant to process establishment, process assessment, and the project life cycle is important.

Basic continuous process improvement service operations are as follows:

- identifying and maintaining baseline of the process improvement plan
- maintaining and reporting historical, technical, and evaluation data for process improvement
- managing changes to the process improvement plan
- maintaining and reporting process measurement data
- tracing assessment reports that compare the actual data with target or baseline data

Automated continuous process improvement support service operations are as follows:

- process improvement plan is baselined
- historical, technical, and evaluation datum is stored and changes are controlled
- continuous process improvement status is stored and changes are controlled

B.4 Quality assurance support services

This section represents a collection of capabilities that provide support to the quality assurance process. The QA process which is normally associated with verification, validation, joint review, and audit is applied to both product and process assurance. A CM service supports the definition, tracking, and performance of these assurance activities and the analysis of their results.

B.4.1 Product assurance

Services concept

Product assurance service should have the following capabilities:

- defining and tracking the performance of quality assurance activities and the analysis of their results
- providing the identification, collection, filing, maintenance, and disposition of product quality records
- establishing resources, schedules, responsibilities for conducting product assurance activity

Basic product assurance service operations are as follows:

- identifying and maintaining the baseline for quality product standard, methodology, procedures, and tools
- managing changes to baselined items
- maintaining and reporting records for quality assurance activities
- tracing analysis data for the compliance with requirements

Automated product assurance support service operations are as follows:

- product quality standard, methodology, procedures and tools are baselined
- changes to baselined items are managed and reported
- link product assurance activity to V&V, joint review, audit, and risk management activity

B.4.2 Process assurance

Services concept

Process assurance service should have the following capabilities:

- assuring the conformance between the contract requirements and the software lifecycle processes
- assuring the conformance between the contract requirements and the internal software engineering practices, development environment, test environment and libraries
- assuring that software process measurements are in accordance with established standards and procedures

Basic process assurance service operations are as follows:

- identifying and the managing baseline of contract requirements, defined software lifecycle processes, internal software engineering practices, development environment, test environment, libraries, standards, and procedures
- managing changes to the above baselines
- maintaining and reporting records of process quality assurance activities

- tracing the analysis data for compliance with requirements

Automated process assurance support service operations are as follows:

- contract requirements, defined software lifecycle processes, internal software engineering practices, development environment, test environment, libraries, standards and procedures are baselined
- changes to the above configuration items are maintained and controlled
- audit data is provided

B.5 Reuse support services

Reuse services represent a collection of capabilities that provide the supports the reuse of project and process assets. Since the reuse process encompasses not only software work products, but also process, standards, and procedures, it is important that all the resources and assets are defined and controlled by CM.

Services concept

Reuse support should have the following capabilities:

- creation
- update
- deletion
- certification
- measurement
- management of process assets (e.g., activities, tasks)

Process assets may range from complete life-cycle process definitions to individual process steps. Process assets may also be the objects that can be versioned.

Basic reuse support service operations are as follows:

- identifying and managing the baseline for asset reuse
- maintaining accessibility to configuration items for reuse
- managing history of reuse
- linking and tracking of configuration items for reuse
- managing changes to process assets

Automated reuse support service operations are as follows:

- information for reuse is identified and baselined
- change to configuration items is controlled and recorded
- traceability among software work products and process assets is provided

Bibliography

- [1] ISO/IEC 9126-1:2001 Software engineering – Product Quality – Part 1: Quality model
- [2] ISO/IEC TR 9126-2:2003 Software engineering – Product quality – Part 2: External metrics
- [3] ISO/IEC 14598 Information technology – Software product evaluation
- [4] ISO/IEC TR 19759:2004 Software engineering – Guide to the software engineering body of knowledge
- [5] ISO/IEC 15504 Information technology – Software process assessment
- [6] ISO/IEC 15940:2006 Information technology – Software engineering environment services
- [7] ISO/IEC 14102:2008- Information technology- Guideline for the evaluation and selection of CASE tools
- [8] ISO/IEC 25000:2005 – Software Engineering - Software product Quality Requirements and Evaluation (SQuaRE) - Guide to SQuaRE
- [9] ISO/IEC 25030:2007 - Software Engineering - Software product Quality Requirements and Evaluation (SQuaRE) - Quality requirements
- [10] ISO 10007:2003, Quality management systems—Guidelines for configuration management
- [11] IEEE 610.12-1990 Standard Glossary of Software Engineering Terminology
- [12] IEEE 828:1990 IEEE Standard for Software Configuration Management Plans
- [13] ANSI/EIA 649-1998 National Consensus Standard for Configuration Management
- [14] Space Project Management. Configuration Management. European Cooperation for Space Standardization. ECSS–M–40a. April 1996
- [15] Space Product Assurance. Software Product Assurance. ECSS–Q–80 A. April 1996

ISO/IEC JTC1/SC7 /N4186

2009-01-05

Document Type	Comments Disposition Report
Title	Comments Disposition Report, N3886, FCD18018 Information Technology — Configuration Management Tool Requirements.
Source	WG4
Project	18018
Status	Final
Reference	N3886, N4011, Resolution 1040, Resolution 1042
Action ID	FYI
Distribution	AG
No. of Pages	15
Note	

Address reply to: ISO/IEC JTC1/SC7 Secretariat
École de technologie supérieure – Department of Software and IT Engineering
1100 Notre Dame Ouest, Montréal, Québec Canada H3C 1K3
secretariat@jtc1-sc7.org

www.jtc1-sc7.org

ISO/IEC JTC1/SC7 4/N-445

2008-12-8

Document Type	Comment Disposition Report
Title	Comment Disposition Report, DTR 18018, Information Technology – Guide for Configuration Management Tool Capabilities
Source	SC7/WG4
Project	
Status	Final
Reference	
Action ID	FYI or ACT
Due Date	
Mailing Date	2008-12-8
Distribution	AG
No. of Pages	14
Note	

Address reply to: ISO/IEC JTC1/SC7 Secretariat
École de technologie supérieure – Department of Software and IT Engineering
1100 Notre Dame Quest, Montréal, Québec Canada H3C 1K3
secretariat@jtc1-sc7.org

www.jtc1-sc7.org

			Balloted	document:	FCD18018 Information Technology — Guide for Configuration Management Tool Capabilities			
					Appr: 13, Appr Comm: 3 (France, Italy, USA), DisAppr:4 (AUS, Czech, JP, UK)			
NB	No.	Category	Clause, Sub-clause	Paragraph, Figure, Table	Comment and rationale	Proposed new text	Outcome	Disposition
JP	1	GE	TITLE		The title does not conform to the dispositon 973 at Moscow plenary.	Fix the title as "Guide for Configuration Management Tool Capabilities"	Accept	Changed 'Guidance' to 'Guide'
JP	2	E	Intro.	2nd para.	"However, this document does not address the capability ..."	"However, these documents do not address the capabilities ..."	Accept	Changed as suggested
JP	3	TH	Intro.	2nd para.	This standard does not define "requirements". "There is a gap between CM process requirements and CM process automation services which ..."	"There is a gap between CM process descriptions and corresponding CM process automation services which ..."	Accept	Changed as suggested
JP	4	TH	Intro.		"Requirements" should be changed to "Capabilities". Agreed at the Montreal meeting.	Change "Requirements" to "Capabilities" or more appropriate terms.	Accept	Changed as suggested
JP	5	E	SCOPE		miss spelling "capabilites"	"This standard provides guidance for the CM tool capabilities"	Accept	Changed as suggested
JP	6	GT	Norm. Ref.		IS12207, 15288, 14102, 14598 are already published or in publishing.	Update references. Move non Normative references to Bibliography. (See IS12207:2008)	Accept	Changed as suggested
JP	7	E	3	Terms and Conditions	Inconsistent use of "." at each sentences.	Conform with the Directives.	Accept	Changed as suggested
JP	8	E	5.2	1st line	There are "this International Standard" and "this Standard" in text.	Use "this International Standard".	Accept	Changed as suggested
JP	9	TH	5.3	2nd line	This standard does not define "requirements".	Delete this line.	Accept in principle	Changed 'requirements' to 'capabilities'
JP	10	TH	6	TITLE	This standard is not intended to make "Requirements of configuration management tools". "Requirements" should be changed to "Capabilities". Agreed at the Montreal meeting.	Change the text to "Capabilities of configuration management tools".	Accept	Changed as suggested
JP	11	GT	6.1, 6.2, 6.3, 6.3.7		There are many "requirements" in text. We agreed that all the "Requirements" should be changed to "Capabilities"at Motreal meeting.	Change "Requirements" to "Capabilities" or more appropriate term.	Accept	Changed as suggested

NB	No.	Category	Clause, Sub-clause	Paragraph, Figure, Table	Comment and rationale	Proposed new text	Outcome	Disposition
JP	12	E	6.2		"requirements" should be changed to "capabilities".	Reword the text to "Rationale of categorization of Configuration Management Tool capabilities", for example.	Accept	Changed as suggested
JP	13	TL	6.2		This part is an informative text and appropriate to Introduction.	Move them to the Introduction.	Accept in principle	Moved the section into 6.1 as a note
JP	14	E	6.3	11th line and after	There are many '&' used in text.	Change all "&" to "and".	Accept	Changed as suggested
JP	15	E	6.3.1~		There are many "CI"s and "CI" may be mislead configuration identification, configuration ID, or check-in.	Change all "CI" to "configuration item" explicitly.	Accept	Changed as suggested
JP	16	GT			"ID" makes confusion, there are "configuration ID", "version ID", "configuration & version ID", "configuration item version", and "ID".	Define them clearly and use them properly.	Accept	Changed as suggested
JP	17	E	6.3, 6.3.X	sub-tiltles of 6.3.x	The text in between 6.3 and 6.3.1 does not fully correpond to sub-clauses of 6.3.x.	The text in between 6.3 and 6.3.1 does not conform with the Directives. Also the text is just an incomplete listing of sub-clauses of 6.3.x.	Accept	Changed the text to be consistent with the directives and sub-clause of 6.3.x
JP	18	TL	6.3.1.2, 6.3.1.3, 6.3.3.4.4, 6.3.3.4.5	"software/system element"	no definition or description of software/system element. What are they?	Define "software/system element" and describe relationship to "configuration item" at first in 6.1. Or use "configuration item" instead.	Accept	Provided a definition; Element that define and prescribe what a software or system is composed of (for example, <u>requirements</u> , <u>design</u> , <u>code</u> , <u>test cases</u> , and <u>version number</u>). An element can contain sub elements, or other software/system elements that are dependent on the top level element.

NB	No.	Category	Clause, Sub-clause	Paragraph, Figure, Table	Comment and rationale	Proposed new text	Outcome	Disposition
JP	19	TL	6.3, 6.3.5.2	"software item"	no definition or description of software item.	Describe "software item" and describe relationship to "configuration item" at first in 6.1. Or use "configuration item" instead	Accept	Changed 'software item' to 'configuration item'
JP	19	E	6.3.3.2.3	1st line	No explanation of "Configuration Control Board (CCB)".	Explain "CCB".	Accept	Inserted the following sentence; The board reviews each proposed change, approve or disapprove it, and if approved, coordinate the change with the affected groups
JP	20	GE	Annex		Wrong Annex style.	Conform with the Directives.	Accept	Changed the format as directed
JP	21	GE	Annex	Table	Wrong table style.	Conform with the Directives.	Accept	Changed as directed
JP	22	E	Annex-A	A.2,A.3	Unnecessary IS12207-15288 cross-refereces in Informatrive Annex.	Delete A.2 and A.3.	Accept	Deleted
JP	23	E	Annex B	"Benefits of configuration management"	Unnecessary Annex B (Benefits of configuration management). It is already explained in caluse 5 (Application of this International Standards) .	Delete Annex B.	Accept	Deleted
JP	24	E	Annex C		This standard is not intended to state any CM services. Also, there are inconsistency in the body text.	Delete Annex C.	Accept in principle	Although defining the CM service is not an intention of this standard, it helps users in evaluating and choosing appropriate CM capabilities for a specific process activity. Therefor left some core parts, as an example.
JP	25	GT			There are many inconsistency and this is still immature as FCD. Make this as a TR.		Accept partially	Inconsistency is eliminated; Will be following a TR track

NB	No.	Category	Clause, Sub-clause	Paragraph, Figure, Table	Comment and rationale	Proposed new text	Outcome	Disposition
GB	1	GT			It is not clear what value this standard offers in addition to the configuration management standards already available. It is like having a 'sister' standard for all the IT standards where a tool or application product could be applied.		Accept in principle	Existing CM standards does not prescribe the capabilities of a tool to be automated. This document focuses only on tool guidance which other standards do not focus.
GB	2	GT			A vendor could use this to say that their tool is compliant with the standard and yet the application in an organization could be non conformant		Accept in principle	This documents will be a TR. The purpose of this standard is not to provide the conformity of standard but to provide the usage guidance
GB	3	GT			There is no mention of 20000 and 19770. The title and applicability statement must be limited to CM tool requirements to exclude 20000 and 19770 so that vendors cannot imply that their tools are wholly suitable for supporting the requirement of 20000 and 19770. Some of the requirements are specific for software and projects. There is little mention of production services, infrastructure and hardware assets.		Accept in principle	This standard is based on CM process described in 12207 and 15288. IT service point of view is out of scope for this standard.
GB	4	GT			Nothing about stakeholder management		Accept in principle	This standard is based on CM process described in 12207 and 15288. Stakeholder management is out of scope for this standard.

NB	No.	Category	Clause, Sub-clause	Paragraph, Figure, Table	Comment and rationale	Proposed new text	Outcome	Disposition
	5	GT			Missing the need to define the scope and requirements for the configuration management tool /application and manage the requirements for different stakeholder use cases and roles.		Accept in principle	This standard is based on CM process described in 12207 and 15288. Stakeholder management is out of scope for this standard.
GB	6	TH			Support for release and deployment management missing.		Accept in principle	This standard is based on CM process described in 12207 and 15288. IT service point of view is out of scope for this standard. The software point of view for release management is also described in 6.2.6
GB	7	TH	3.8		Definition of CM Services - "Abstract description of work done by CM tools. A service is self contained, coherent, discrete, and may be composed of other services." The definition limits the use of the word "service" that will impact harmonisation with 20000.		Accept in principle	The definition is quoted from the IS 15940 which is one of the key reference standards. IT service point of view is out of scope for this standard.
GB	8	TH	3.8 & 3.9		CM service / CM tool. The CM could mean change management or capacity management - use of abbreviations is confusion		Accept	The definition of CM has been explained in the same section.
GB	9	TH			No mention of security and access management requirements that would be required for any configuration management tool e.g. access limited by role and security requirements		Accept partially	See the description in 6.2.3.3 under the configuration control section, 6.2.3
GB	10	TH			Many core requirements missing e.g. control over labelling, versioning, branching and baselining of configuration items		Accept partially	See 6.2.1, 6.2.2, 6.2.3.4

NB	No.	Category	Clause, Sub-clause	Paragraph, Figure, Table	Comment and rationale	Proposed new text	Outcome	Disposition
GB	11	TH	1. Scope		Lack of clarity about the scope of the document, and the impact it would have on other areas without that clarification	The title of the standard and the scope need to be clearly defined - for example Title change to "Software Development Lifecycle Configuration Management Tools" and scope re-written to reflect this.	accept	Clarified the scope of this document as directed
AU	1	TE	3.2		Clarify relationship between baselines and CIs and/or versions	"A baseline is a version of a configuration item for a specification or product that ..."	Accept partially	Since the configuration itself can be a baseline, the definition uses configuration, instead of configuration item, and is changed as follows; a version of a configuration for a specification or product....
AU	2	TE	3.4		Archiving is not normally regarded as a build activity. The build process is also not always about executable systems that must be compiled – don't restrict the definition to executable/compiled outputs.	"The process of generating a derived system from source versions. The build needs to transform and bring together the various versions in the correct order."	Accept partially	The term was harmonized in accordance to the one defined in other standards such as 12207 and 15288
AU	3	TE	3.10	definition	"actual" difference is not specified – lines? characters? bytes? What about binaries? Also, a delta isn't a difference between "each" version – just between 2 specific versions.	"The difference between two versions."	Accept	Changed as suggested

NB	No.	Category	Clause, Sub-clause	Paragraph, Figure, Table	Comment and rationale	Proposed new text	Outcome	Disposition
AU	4	TE	3.10	Note	Recording differences “in discrete files” is a solution-specific suggestion – should not be in the standard. Also, deltas do not always “reduce data redundancy”.	(delete the whole note)	Accept	Changed as suggested
AU	5	TE	3.12		Must versions be immutable? If so, say so in the definition, and delete the note (you can’t “modify” a version to create a new version – either the version changes or it does not). If not, delete the somewhat confusing note.	(delete the whole note)	Accept in principle	Yes, versions are immutable. The definition has been harmonized with 12207.
AU	6	TE	6.1	1st para	A CM tool does NOT provide a disciplined approach in and of itself.	“A CM tool can provide automated assistance for CM activities: ...”	Accept	Changed as suggested
AU	7	TE	6.3	1st line	Many of the capabilities listed do not require a tool to be “enabled”, but could nonetheless be “supported” by a tool.	“The CM tool should provide capabilities supporting the following CM activities:”	Accept	Changed as suggested
AU	8	TE	6.3	baselining	Baselines are just versions of CIs. All parts of the category of baselining should be in other sections.	move 1st three points within baselining to “Configuration Identification” above move “controlling access..” point to “Configuration Control” below delete whole baselining category	Accept partially	Not all configuration items identified becomes a baseline. Thus baselining is different from configuration identification activity. However, controlling access to baseline can be moved to configuration control section.

NB	No.	Category	Clause, Sub-clause	Paragraph, Figure, Table	Comment and rationale	Proposed new text	Outcome	Disposition
AU	9	TE	6.3	configuration control	A CM tool can not itself provide a capability to: - analyse or evaluate changes (requires human expertise) - approve or disapprove requests (requires human authority) - implement changes (CM tools are not automatic programmers!)	Delete points 2-4, or significantly re-cast them to make it apparent that a CM tool can only hope to support the management of these activities.	Accept partially	The bulleted items generally describes the tool supports for change control. Surely, analysing or evaluating changes can not be done by the tool. But the tool can support these activities by providing templates or checklists. What a CM tool can do is described in the following sub-sections.
AU	10	ED	6.3	disposition	disposition of what?	“communicating the disposition of change requests”	Accept	Recasted the bullet items to be consistent with the headings in sub-sequent sections
AU	11	ED	6.3	audit trail	“audit trail are” – wrong part of speech	“managing an audit trail to trace the...”	Accept	Recasted the bullet items to be consistent with the headings in sub-sequent sections
AU	12	TE	6.3	status accounting	“comparisons of releases” – by what metric/feature?	delete “, comparisons of releases”	Accept	Recasted the bullet items to be consistent with the headings in sub-sequent sections
AU	13	TE	6.3	other	“integration with other tools” – no criteria/boundary for integration given	delete “the integration with other tools” OR expand to make a claim about what other tool integration should be possible	Accept in principle	The criteria/boundary for integration is explained in the sub-sequent section. 6.3 only represents the topics

NB	No.	Category	Clause, Sub-clause	Paragraph, Figure, Table	Comment and rationale	Proposed new text	Outcome	Disposition
AU	14	TE	6.3	other	The standard has studiously ignored discussing which technical/commercial contexts it might be relevant to – why start now? There’s no reason to think that web site development wouldn’t already be within the scope of the standard.	delete “, and web site development”	Accept	Deleted
AU	15	TE	6.3.1.2		Is the “name of CI” the same as the “unique identifier”?	delete “generating a unique identifier... each item”	Accept in principle	The "name of CI" is a characteristic that the tool should capture,. This chatacteristics is supported by providing a unique identifier.
AU	16	TE	6.3.1.2	“e.g. alphanumeric”	“e.g. alphanumeric” is at the same time over-constraining and also meaningless	delete “e.g. alphanumeric”	Accept in principle	It is just an example. To clarify the meaning, inserted 'alphabetic' too.
AU	17	TE	6.3.1.3	“CI sources”	what are “CI sources”?	change to “tracing between CIs”	Accept	Changed as suggested
AU	18	TE	6.3.1.4	“conditions or events”	not enough to just “enable” the change in status – also need to make sure it happens. Also, a status change may or may not be thought of as an “advancement” – e.g. when a system tester fails a change back to be re-implemented	change to “defining the conditions or events that enable or enact the change of CI status”	Accept	Changed as suggested
AU	19	TE	6.3.2		A baseline is just a version of a CI. These capabilities are (or should be) dealt with in other sections.	delete whole section	OBE at AU 8	
AU	20		6.3.7.1	flexible	“flexible” is meaningless	delete “flexible”	Accept	Changed as suggested
AU	21		6.3.7.1	environment services	the environment services exist independently of the CM tool/this standard – all we can hope for is to be able to use them	“enable the use of multiple implementations of environment services”	Accept	Changed as suggested

NB	No.	Category	Clause, Sub-clause	Paragraph, Figure, Table	Comment and rationale	Proposed new text	Outcome	Disposition
AU	22		6.3.7.1	CASE	It's unreasonable to suggest that "all the CM functions" could be performed within the CASE environment – such tools are usually targeted at specific functions and do not support CM administrative functions (e.g. CM audit, CM planning, etc)	change to "sharing the workspace with CASE tools so that users can perform relevant CM functions within the shared workspace"	Accept	Changed as suggested
AU	23	ED	6.3.7.2		1st point is grammatically incorrect	change to "provide members of the group the ability to receive ... the group"	Accept in principle	By agreeing on other NB's comments that this section goes beyond of the CM specific, this section is deleted.
AU	24	TE	6.3.7.2		group members can receive information from other regardless of the CM tool (e.g. email/phone/etc) – is there a purpose to this requirement (e.g. the nature of the information?) that somehow relates to CM	delete "provide members of the group ... the group"	OBE AU 23	
AU	25	ED	6.3.7.2	more than one user	grammatical problem - quantifier ordering	"provide the ability for multiple users to modify the same files ... interactively"	OBE AU 23	
AU	26	TE	6.3.7.2	merge	Modifying the same files is not always possible (e.g. with binaries) or desirable (e.g. under some organizational branching policies/patterns)		OBE AU 23	

NB	No.	Category	Clause, Sub-clause	Paragraph, Figure, Table	Comment and rationale	Proposed new text	Outcome	Disposition
CZ	1	GE			<p>The standard refers for the standards of the ISO/IEC 9126 and ISO/IEC 14598 series for software quality model and software quality evaluation. These standards are behind the times and now are replaced by the standards of ISO/IEC 250xx series developed in SQuaRE project. It is necessary to wait for the finishing of corresponding standard of this series and arrange the terminology to be consistent with the new standards.</p> <p>It is not reasonable to downline the different terminology. For example attribute is defined as a measurable property of an entity. It is diametrically different concept as the concept in the draft: “attributes are entities that define properties of ... “. The new concept is fully addle-brained.</p>		Accept in principle	There are a set of definitions on 'attribute' in ISO/IEC JTC1/SC7 standards (NP 24765 Systems and software engineering - vocabulary).
IT	1	GE			Not clear what the title is: CM Tool Requirements or Guidance to CM Tool Capabilities ? It seems from the text that it should be guidance and not requirements.		Accept	It is not about 'requirement'. Changed the title to 'Guide'
IT	2	GE			In many places there is only a reference to software. CM applies also to system.	Check all occurrences of software and add system also.	Accept	Changed as suggested

NB	No.	Category	Clause, Sub-clause	Paragraph, Figure, Table	Comment and rationale	Proposed new text	Outcome	Disposition
IT	3	TH	Introduction	2nd parag. 2nd sentence	This sentence says that the document does not address the capability that a CM tool user can expect...etc. It does not seem to make sense because it seems the exact contrary.		Accept in principle	Those existing documents only focuses on CM process activities in a higher level of abstraction , not on CM tool capabilities in detail. Not all the CM process activities can be supported by the tool. This document focuses on what CM tool capabilities support the CM process activities.
IT	4	TH	Introduction	3rd parag. Bullet 2)	Eliminate the reference to requirements, because the documents contains guidance. Furthermore 12207 does not specify requirements to be met by tools.	2) Provide guidance for tool manufactures in implementing a minimum set of capabilities. The <u>tool capabilities</u> defined in this standard are linked to IS 12207 and IS 15288 and will provide tool manufacturers guidance as to what characteristics their tools <u>should meet to support these</u> ISO/IEC standards.	Accept	Changed as suggested
IT	5	E	Normative Ref.		Update publication year of 12207 and 15288. They have just been republished in 2008.	Check also Annex A	Accept	Changed as suggested
IT	6	E	6.2	Title	rather than "rationale" it seems like a bibliography.	6.2 <u>Bibliography used for</u> configuration management tool requirements categorization	OBE JP 13	
IT	7	E	6.3	Title	Eliminate the word description. Not needed. Change requirements in capabilities	6.3 Configuration management tool capabilities	Accept	Changed as suggested

NB	No.	Category	Clause, Sub-clause	Paragraph, Figure, Table	Comment and rationale	Proposed new text	Outcome	Disposition
FR	1	GE	6.1 6.3.1.2 6.3.3.3.4	First sentence First item, second line of page 10 First line	Is the target of the standard to manage SW or systems configuration? As a consequence, is the elementary component to consider a configuration item, a file or a code source?	Please clarify the ambiguity.	OBE IT 2	
FR	2	TE			There is nothing dealing derived items, eg. links between source code and executable. That point is a strong feature of tools such as Clearcase and Clearmake.	Add a paragraph on derived items.	Accept	Included in 6.4.4
FR	3	TE	6.3.5.4		In the standard, it is said that tools for configuration management should include project management features. Project management is not, for our point of view, a mandatory feature for configuration management tools.	Amend that feature in the standard, stating that project management may be an optional feature for configuration management tools.	Accept	Changed as suggested.
FR	4	TE			Traceability is mentioned in many parts of the standard, without some clear definition.	There are different kinds of traceability: dependence between items (derivation), between specifications and tests (see § 6.3.2.4). That last kind of traceability should rather be included in other kinds of tools, like requirement engineering (Doors) or tests.	Accept	Defined the traceability in 3.14
FR	5	TE	6.3.1.3		It is not clear that CI are nodes of a PBS.	Please clarify this point.	Accept in principle	CIs are not necessarily nodes.
FR	6	TE	6.3.3.2.1	First item	“change effort” should be known after having proceeded analysis changes (6.3.3.2.2)	“change effort” should be in 6.3.3.2.3	Accept	Changed as suggested.
FR	7	TE	6.3.3.4	Par 2	It seems that a CI could be modified at any time without creating a new version of it. This statement is very strange for us, as a version should be stable, hence it should be not modified. Any modification would lead to the creation of a new version.	Please amend that point.	Accept	Changed the sentence by stating that every modification to a version results in a new version

NB	No.	Category	Clause, Sub-clause	Paragraph, Figure, Table	Comment and rationale	Proposed new text	Outcome	Disposition
USA	1	TH	A.1	12207 Section and Table	The recently approved version of 12207 has separate processes for system-level Configuration Management and Software Configuration Management.	Update the reference to ISO/IEC 12207:2008. List both the system-level CM process and the Software CM process. Ensure that the lists of activities are correct.	OBE JP 6	
USA	2	TL	A.1	15288 Section and Table	The revision to ISO/IEC 15288 was recently approved. This should refer to the new version,	Update the reference to ISO/IEC 15288:2008. Ensure that the list of activities is correct.	OBE JP 6	
USA	3	TH	A.2	Table	It appears that this table is intended to list relevant processes of 12207. The column header is incorrect and an obsolete version of 12207 has been used for the content of the table.	Change the table to show the relevant processes in ISO/IEC 12207:2008 and change the column header to read "processes" instead of "activities".	OBE JP 22	
USA	4	TL	A.3	Table	The items listed for 15288 are not activities. These are processes per 15288:2008.	Change the left column title of the table to read "processes" instead of "activities".	OBE JP 22	

ISO/IEC JTC1/SC7 /N4198

2009-01-19

Document Type	Abstract
Title	Abstract, DTR18018 Information Technology — Configuration Management Tool Requirements.
Source	WG4
Project	18018
Status	Final
Reference	N3886, N4011, Resolution 1040, Resolution 1042, N4186, N4187
Action ID	FYI or ACT
Distribution	AG
No. of Pages	1
Note	

Address reply to: ISO/IEC JTC1/SC7 Secretariat
École de technologie supérieure – Department of Software and IT Engineering
1100 Notre Dame Ouest, Montréal, Québec Canada H3C 1K3
secretariat@jtc1-sc7.org

www.jtc1-sc7.org

ISO/IEC DTR 18018 Abstract

Configuration Management (CM) is a process central to the software engineering life cycle. CM has been established as an ISO/IEC standard life cycle process in ISO/IEC 12207:2008 Information Technology – Software Life Cycle processes and ISO/IEC 15288: 2008 Information Technology - System Life Cycle processes.

IS 12207 and IS 15288 describe a comprehensive set of processes, activities and tasks to be performed when acquiring or developing software. However, these documents do not address the capabilities that a CM tool user can expect from a tool in order to support the CM process and other software engineering life cycle activities. There is a gap between CM process descriptions and corresponding CM process automation services which affects both tool users and tool suppliers.

The outcomes of this technical report are to:

- 1) Provide guidance in the evaluation and selection for CM tools during acquisition. CM tool evaluation by prospective users can be complex, time consuming, and expensive. This standard helps to characterize what a CM tool can do or cannot do in the CM process.
- 2) Provide guidance for tool manufactures in implementing a minimum set of capabilities. The capabilities defined in this standard are linked to IS 12207 and IS 15288, and will provide tool manufacturers guidance as to what characteristics their tools should support to meet these ISO/IEC standards.

ISO/IEC DTR 18018 Keywords

Configuration Management (CM), CM tool, CM tool capabilities, CM process, Configuration identification, Configuration baselining, Configuration control, Configuration status accounting, Configuration auditing, Release management

G8 Explanatory Report

EXPLANATORY REPORT	ISO 25021 - Software Engineering: Software product Quality Requirements and Evaluation (SQuaRE) - Quality measure elements
ISO/IEC JTC 1/SC 7 N 3697	
Will supersede:	Secretariat: National Body (SCC)

This form should be sent to ITTF, together with the committee draft, by the secretariat of the joint technical committee or sub-committee concerned.

The accompanying document is submitted for circulation to member body vote as an DTR, following consensus of the P-members of the committee obtained on:

2005-05-20

by postal ballot initiated on: 2005-02-20

P-members in favour: 9

P-members voting against: 2

P-members abstaining: 3

P-members who did not vote: 14

Remarks:

Project: JTC 1

I hereby confirm that this draft meets the requirements of part 3 of the IEC/ISO Directives

Date: 2007-02-05

Name and signature of secretary: Witold Suryń