INTERNATIONAL STANDARD

ISO/IEC/IEEE 26531

First edition
2015-05-15

Systems and software engineering — Content management for product life-cycle, user and service management documentation

Ingénierie des systèmes et du logiciel — Gestion de contenu relatif a la documentation du cycle de vie du produit, de l'utilisateur, et de la gestion de service

Reference number
ISO/IEC/IEEE 26531:2015(E)

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

IEC Central Office
3, rue de Varembé
CH-1211 Geneva 20
Switzerland
E-mail inmail@iec.ch
Web www.iec.ch

Institute of Electrical and Electronics Engineers, Inc.
3 Park Avenue, New York
NY 10016-5997, USA
E-mail stds.ipr@ieee.org
Web www.ieee.org

Published in Switzerland

Contents

Page

List of Illustrations

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.

IEEE Standards documents are developed within the IEEE Societies and the Standards Coordinating Committees of the IEEE Standards Association (IEEE-SA) Standards Board. The IEEE develops its standards through a consensus development process, approved by the American National Standards Institute, which brings together volunteers representing varied viewpoints and interests to achieve the final product. Volunteers are not necessarily members of the Institute and serve without compensation. While the IEEE administers the process and establishes rules to promote fairness in the consensus development process, the IEEE does not independently evaluate, test, or verify the accuracy of any of the information contained in its standards.

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

The main task of ISO/IEC JTC 1 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.

Attention is called to the possibility that implementation of this standard may require the use of subject matter covered by patent rights. By publication of this standard, no position is taken with respect to the existence or validity of any patent rights in connection therewith. ISO/IEEE is not responsible for identifying essential patents or patent claims for which a license may be required, for conducting inquiries into the legal validity or scope of patents or patent claims or determining whether any licensing terms or conditions provided in connection with submission of a Letter of Assurance or a Patent Statement and Licensing Declaration Form, if any, or in any licensing agreements are reasonable or non-discriminatory. Users of this standard are expressly advised that determination of the validity of any patent rights, and the risk of infringement of such rights, is entirely their own responsibility. Further information may be obtained from ISO or the IEEE Standards Association.

ISO/IEC/IEEE 26531 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information Technology*, Subcommittee SC 7, *Systems and Software Engineering* in cooperation with the Systems and Software Engineering Standards Committee of the IEEE Computer Society, under the Partner Standards Development Organization cooperation agreement between ISO and IEEE.

Introduction

This International Standard was developed to assist users of ISO/IEC/IEEE 15288:2008, *Systems and software engineering – System life cycle processes*, ISO/IEC/IEEE 12207:2008, *Systems and software engineering – Software life cycle processes*, or ISO/IEC 20000-1:2011 (IEEE Std 20000-1-2013), *Information technology – Service management – Part 1: Service management system requirements*, in the management of the content used in product life-cycle, user, and service management documentation. The accurate description of the requirements for content management helps documentation meet the needs of its users and be efficiently produced.

This International Standard is independent of the software tools that may be used to manage documentation content and applies to both printed documentation and on-screen documentation.

Content management allows an organization to control the storage and retrieval of content objects, track content revisions, maintain a content audit trail, and enable a collaborative environment. Component content management supports the reuse of content objects among deliverables and supports multiple deliverable formats.

A consequence of content management is increased collaboration on content development across the enterprise. Technical authors, instructional designers, support staff, and others may develop a body of content together that is written once and supports many needs.

Documentation is often regarded as something done after the software has been implemented. However, for high-quality software documentation, its development should be regarded as an integral part of the software life cycle. In fact, quality documentation or information management services are important enough to require specific planning.

This International Standard is consistent with ISO/IEC/IEEE 15288, *System and Software Engineering – System life cycle processes,* and ISO/IEC/IEEE 12207, *Systems and software engineering – Software life cycle processes*, as an implementation of the Information Management Process. This standard is not a management system standard.

This International Standard is intended for use in all types of organizations, whether they have a dedicated documentation department or not. It may be used as a basis for local standards and procedures. Readers are assumed to have experience or knowledge of general processes for information management, project management, and document development.

This International Standard is intended for managing technical content which is included in:

- User information such as topic collections, manuals, guides, multimedia, user assistance displayed with software, style guides, and other content that supports the effective use of a system or software product.
- Product life cycle information such as design documents, use cases, personas, project management plans, feature requests, and testing plans.
- Service management items such as service level agreements, records, policies, procedures, and other documents.

The order of clauses in this International Standard does not imply that the content management activities should be performed in this order, nor that documentation should be developed in this order or presented to the organization in this order.

In each clause, the requirements are independent of media and document creation and management specifications.

Systems and software engineering — Content management for product life-cycle, user and service management documentation

1 Scope

This standard states requirements for efficient development and management of content produced

- throughout the life-cycle of a system and software product;
- for the provision of user documentation for systems and software;
- for the management of IT services.

This standard is independent of the tools, protocols, and systems used for content management. It does not address configuration management of software assets.

The content to be managed with this standard includes

- user information such as topic collections, manuals, guides, embedded user assistance, style guides, videos and other media, and other content that supports the effective use of a system or software product;
- product life cycle information such as design documents, use cases, personas, project management plans, feature requests, models, scripts, testing plans, test scripts, defect reports;
- service management items such as service-level agreements, records, policies, procedures, and other documents.

The purpose of this standard is to define a process for content management and the requirements of a component content management system through which content is gathered, managed, and published, including the requirements of a system that is supported by an electronic database. Such a database should support documents or topics and content units that may be assembled to produce complete documents for print, electronic output, or content collections published through electronic media. This database is defined as a Component Content Management System (CCMS), which differs from a document management system. The objective of component content management is to create content objects once and use them through linking mechanisms in multiple output formats, including but not limited to documents.

The intended users of this International Standard are managers and developers of information (technical documentation) and acquirers and suppliers of content management systems. Any organization that develops content, regardless of size, can benefit from maintaining an effective content management solution and following best practices for the development and management of technical content.

Systems conforming to this standard can fulfill business needs for content development and management, especially the need for a single source of authoritative information. Content objects that are unique and are maintained as independent database objects are efficient to review, approve, and update; may be combined to produce multiple deliverables; and are cost-effective to translate.

This standard is not a management system standard.

The content management process presented in clauses 6 through 11 of this International Standard is a specialization (lower-level process) of the Information Management process required in ISO/IEC/IEEE 15288:2008 and ISO/IEC/IEEE 12207:2008.

2 Conformance

This International Standard may be used as a conformance or a guidance document for projects and organizations claiming conformance to ISO/IEC/IEEE 15288:2008, *Systems and software engineering – System life cycle processes*, or ISO/IEC/IEEE 12207:2008 *Systems and software engineering – Software life cycle processes*.

Throughout this International Standard, "shall" is used to express a provision that is binding, "should" to express a recommendation among other possibilities, and "may" to indicate a course of action permissible within the limits of this International Standard.

Use of the nomenclature of this International Standard for the parts of documentation, such as topics, content units, modules, is not required to claim conformance.

This International Standard may be included or referenced in contracts or similar agreements when the parties (called the acquirer and the supplier) agree that the supplier will deliver services and systems in accordance with the standard. This International Standard may also be adopted as an in-house standard by a project or organization that decides to acquire documentation from another part of the organization in accordance with the standard.

3 Normative references

There are no Normative References applicable to this standard.

4 Terms and definitions

For the purposes of this standard, the terms and definitions given in ISO/IEC/IEEE 24765, available at www.computer.org/sevocab and the following apply.

4.1
application programming interface
software component that allows software applications to communicate with one another

4.2
branching
method of development in which a set of components is duplicated so the components may be modified in parallel and optionally synchronized at a later time

4.3
component
an object with a discrete information type that is stored in the CCMS, such as a topic, prerequisite, section, image, or video

4.4
component content management system
content management system that supports the entire document- or information-development life cycle from authoring through review and publishing, including the reuse of modular content

NOTE In case the modular content is XML-based, the individual XML elements available for management are defined by the XML schema or DTD. This standard is protocol-independent and it is not necessary to specify numerous markup languages.

4.5
component dependencies
all the components that are linked to, directly or indirectly, from a single parent component

EXAMPLE: The relationship of a source hazard statement to its insertion in a topic at time of publication.

4.6
content object
content that is encompassed by an XML element

4.7
content type
reusable definition of settings for storage, metadata, workflow, and behaviour that is assigned to a component

4.8
content unit
identifiable and manageable part of larger information objects

NOTE The individual content units available for management are typically defined by an XML schema or DTD.

4.9
content management system
system that supports authoring, storing, translating, and publishing content

cf. document management system

4.10
customization
modification of a document type definition to add new structures or change the document type definition in a way that is not compatible with a previous structure

4.11
dependency export
operation in which a component and all of its dependencies are exported from the CCMS as a single process

4.12
document management system
system that supports the storage, retrieval, versioning, and manipulation of whole documents, images, and other media

cf. content management system

4.13
document type definition
template for the structure, content, and semantics of XML documents

4.14
documentation
any written or pictorial information describing, defining, specifying, reporting, or certifying activities, requirements, procedures, or results

[IEEE Std 829™-2008]

4.15
effectiveness
accuracy and completeness with which users achieve specified goals

[ISO/IEC 25062:2006]

4.16
EXtensible Markup Language
license-free and platform-independent markup language that carries rules for generating text formats that contain structured data

[ISO/IEC 19770-2:2009]

4.17
EXtensible Style Language Transformations
language for transforming XML documents into other document types such as PDF or HTML

4.18
faceted search
progressive search that allows users to narrow the results by selecting values for one or more attributes

4.19
[CCMS] framework
essential data structures, operations, and rules that form the foundation from which all other features of the CCMS are built

4.20
Hypertext Markup Language
language for creating web pages

4.21
Hypertext Transfer Protocol
application-level protocol for distributed, collaborative, hypermedia information systems

4.22
information item
separately identifiable body of information that is produced, stored, and delivered for human use

[ISO/IEC/IEEE 15289:2015]

NOTE An information item can be produced in several versions during a project life cycle.

4.23
information type
class of topics that addresses a particular user question

EXAMPLE An information type that answers the question “how do I …” is called a task information type.

4.24
Levenshtein distance
measure of the difference between two character sequences based on the minimum number of single character edits (insertion, deletion, or substitution) needed to convert one word to the other

4.25
link
part of a computer program, often a single instruction or address, that passes control and parameters between separate modules of the program

4.26
object
encapsulation of content units in a CCMS

4.27
publishing pipeline
series of defined processing steps that are connected to transform content from its source format into a final deliverable format

4.28
regular expression (Regex)
string of characters that allows patterns to be used to match search results

NOTE Patterns may dictate that matches must start or end with specific sequences of characters or allow the use of wildcards to match any characters in a sequence.

EXAMPLE

^admin* - Find all matches that start with 'admin' and contain any sequence of characters afterwards

\d{5}$ - Find all matches that end with the number 5

^[0-9()-]+$ - Find matches that contain a 10-digit phone number

4.29
semantic label
tag that describes the content rather than its format

EXAMPLE A semantic label such as prerequisite describes the content as a pre-requisite to the following task information. In contrast, a format label simply describes the content as a paragraph or a list.

4.30
specialization
specification of targeted document type definitions that share the common output transformations and design rules developed for more general types and domains

4.31
structured authoring
development of content elements including metadata in specified templates

NOTE In structured authoring, content elements are labeled according to the nature of the content they contain. Structured authoring also permits quasi-semantic labeling, such as Heading1 or NestedList, to indicate the hierarchical position and function of a content element.

4.32
taxonomy
scheme that partitions a body of knowledge and defines the relationships among the parts

4.33
topic
standalone item of information that answers a single user question

4.34
XML Schema definition
XML-based language that specifies a set of rules and structure for the creation of XML documents

[ISO/IEC 19770-2:2009]

5 Abbreviated terms

API Application Programming Interface

BMP Bitmap image file

CCMS Component Content Management System

DITA Darwin Information Typing Architecture

DMS Document Management System

DTD Document Type Definition

ECMS Enterprise Content Management System

GIF Graphics Interchange Format

HTML Hypertext Markup Language

HTTP Hypertext Transfer Protocol

JPEG Joint Photographic Expert Group

LCMS Learning Content Management System

QTFF QuickTime File Format (abbreviated as.mov)

MP3/MP4 See MPEG

MPEG Moving Picture Experts Group

PDF Portable Document Format

PNG Portable Networks Graphics

SCMS Source Control Content Management System

SME Subject Matter Expert

WAV Waveform Audio File Format

WMV Windows Media Video

XLIFF XML Localisation Interchange File Format

XML Extensible Markup Language

XSLT EXtensible Style Language Transformations

6 Content management process

This clause describes the activities of a process through which content is managed from initiation through approval and publishing.

Content management activities described in this standard are as follows:

- Project Initiation
  - Developing a business case for managing content
  - Defining requirements for a CCMS
- Project Plan
  - Information model
  - Authoring guidelines
  - Reuse strategy
  - Metadata schema
  - Workflow specification
  - Schedule of activities, deliverables, and responsibilities
  - Training plan
  - Stylesheet development

  - Pilot project specification
  - Organization rollout
- Information Development
  - Content conversion
  - Content authoring
- Management and control
  - Quality management
  - Review and approval
  - Search and retrieval
  - Localization and translation
  - Content deletion
  - Content archiving
- Publication
  - Release management
  - Publication of content

These can be represented as shown in Figure 1.

The details of each activity in the content management process are discussed in separate clauses.

NOTE ISO 21500, Guidance on project management, can be consulted for managing a content management project.

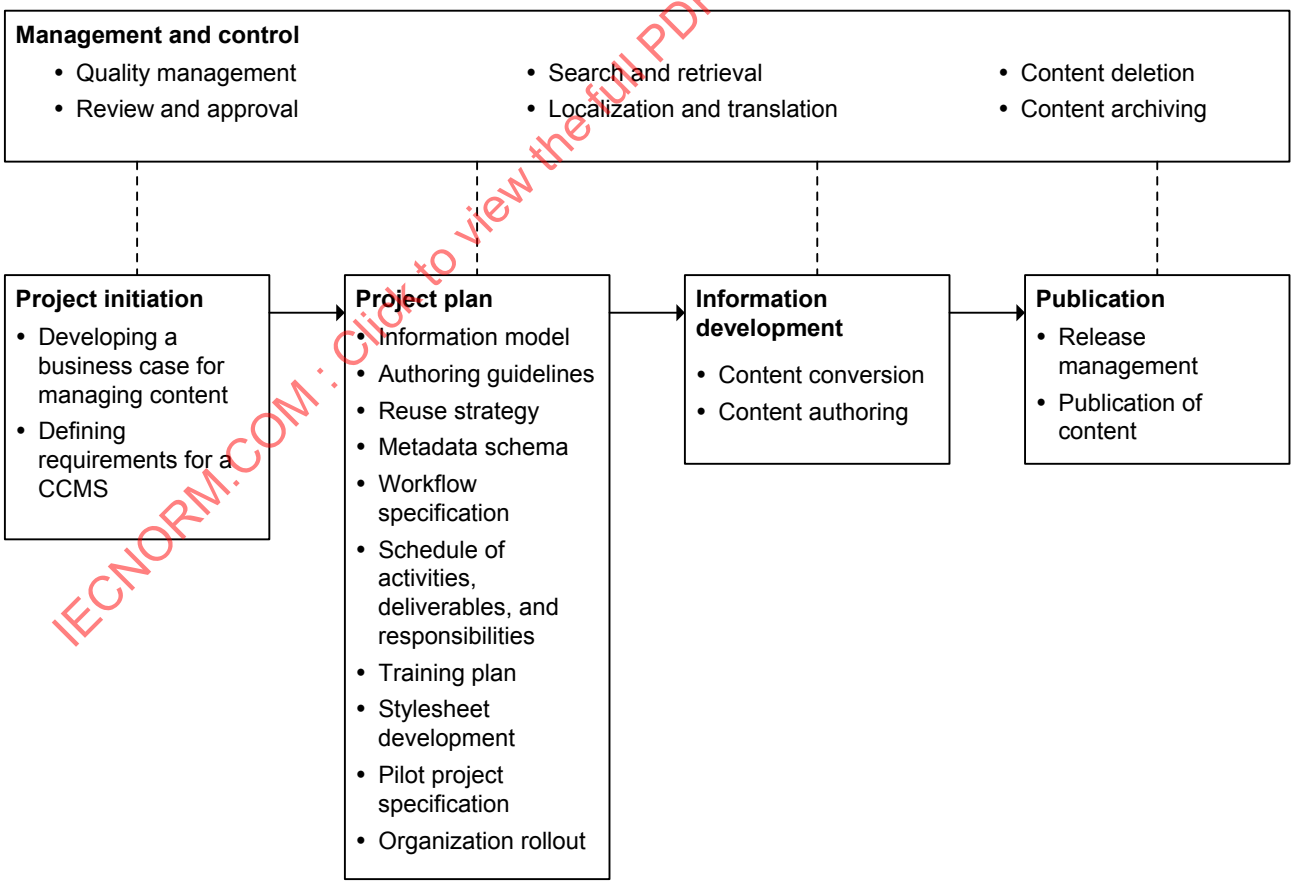

Figure 1 — Content Management process and activities

7 Content management project initiation

7.1 Developing a business case

The organization shall develop a business case to support the development and implementation of a content management solution.

In developing a business case for content management, an organization shall evaluate the needs of the organization, as follows:

- Analyze the current state.
- Identify potential customer benefits.
- Identify cost reduction opportunities.
- Calculate acquisition and implementation costs, including both technology and personnel costs.
- Calculate the Return on Investment.
- State the risks of the project.

The development of a business case may be necessary to obtain support and funding for a content management solution. Establishing the business benefits of content management in an organization includes carefully evaluating both immediate and long-terms gains and evaluating enterprise and departmental solutions. Understanding the present situation assists in developing a vision of why and how that situation should be changed.

Content management allows an organization to control the storage and retrieval of content objects, track content revisions, maintain a content audit trail, and enable a collaborative environment. Component content management supports the reuse of content objects among deliverables and supports multiple deliverable formats.

A consequence of content management is increased collaboration on content development across the enterprise. Technical authors, instructional designers, support staff, and others may develop a body of content together that is written once and supports multiple needs.

By implementing a CCMS and activities to support that system, organizations can experience the following benefits:

- Standardized methods for creating content
- Standardized levels of content quality
- Avoidance of duplicated or near-duplicated content
- Reduced maintenance of content objects
- Greatly improved access and successful search to content objects with appropriate permissions
- Improved ability to find content objects
- Reduced storage and maintenance requirements through consolidation of content in a unified system
- Traceability of review and approval cycles and change management
- Reduced development costs in source and target languages

- Ability to mitigate legal and compliance risk through a single source of authoritative content
- Ability to deliver content in multiple formats according to current and future standards

For more detailed information about developing a business case, see Annex A.

Content management, if well controlled and managed, increases staff productivity and reduces development and publishing costs. Productivity increases are achieved by implementing processes that support structured authoring and content reuse. Structured authoring implies that content is authored according to specified templates, depending upon the type of content to be developed. Content reuse implies that content objects are stored and maintained uniquely, decreasing the cost of updating and translating the same content multiple times.

Content reuse also implies that a content object is a single source of authoritative content, ensuring that the same information and the exact message appears in every instance of output, decreasing the liability associated with incorrect information. Development costs are reduced because existing content is available to authors through search and retrieval, implying that specific information is created and updated only once. Publishing costs are reduced because content objects may be published in more than one format through the use of automated publishing routines.

If content is developed using XML Document Type Definitions (DTDs) or schemas, the cost of formatting that content in multiple languages is eliminated once appropriate style sheets are in place. For those organizations that are required by regulatory bodies or by customer requirements to publish content in multiple languages, CCMSs and processes enable significantly decreased time and cost. CCMSs that enable rigorous linking between content objects also support links between source and target language content objects. Once a body of content has been translated, CCMSs establish links between source and target language content objects so that only new or revised content need be translated. Previously translated content need not be sent again for translation, reducing not only translation costs but also administrative costs in handling already translated content.

7.2 Defining requirements for a CCMS

The business case identifies concerns to be addressed in the explicit definition of requirements for a CCMS.

7.2.1 Requirements definition

The development of a detailed requirements definition enables an organization not only to specify its requirements internally but also carefully evaluate competing products. A well-structured requirements definition focuses on four key areas:

- Output
- Storage and retrieval
- Assembly and linking
- Authoring and workflow

Clause 12 presents the CCMS requirements.

7.2.2 Output requirements

Output requirements shall specify the content delivery mechanisms that make information available to the user community, both internally and externally.

Translation requirements, including the languages and character sets (fonts) required, shall be specified.

An organization may need to target different versions of information to different groups, based on internal needs, market segments, specific customers, and others. Information may need to encompass multiple

versions of a collection of products, where much of the information is shared and some of the information is varied depending on product type or model. Information may be output in multiple media, including print, PDF, embedded user assistance, tablet, electronic applications (apps), e-learning, and others. Information may vary depending on the location or language of the users.

Output requirements should include the potential use of topic- and publication-specific metadata to enable faceted search mechanisms.

Requirements for automatically creating translation packages for transmittal to localization service providers should be specified, including the use of the XLIFF standard (see OASIS standards names) for the transmission of unified data files for translation.

NOTE: The OASIS XML Localization Interchange File Format (XLIFF) provides such a specification at https://www.oasis-open.org/committees/tc_home.php?wg_abbrev=xliff

7.2.3 Storage and retrieval requirements

In preparing storage and retrieval requirements, organizations shall identify how each group using the CCMS will access its content. Storage requirements shall include the ability to maintain links between content objects.

Requirements shall be stated for the storage and retrieval of translated content, especially the requirement to maintain synchronicity between source and target languages to reduce translation costs.

Retrieval requirements shall specify how the CCMS enables internal users to search for and find information.

Various CCMSs support either folder-like structures for navigation to content or search mechanisms that enable users to identify specific content using metadata-enabled search.

Organizations may need to store multiple content types, including images, video, and voice, and entire document constructs as well as storage of individual components that will be assembled into larger constructs on output.

Search mechanisms may be focused on full-text search, metadata-based faceted search, XML-element enabled search, or a combination of these mechanisms.

7.2.4 Assembly and linking requirements

In preparing assembly and linking requirements, organizations shall specify how they expect to collect components of content for assembling into final deliverables, including producing full and partial draft content for reviews.

In an XML environment, linking requirements are especially important because final deliverables are produced through elaborate linking mechanisms. Topics may be assembled into chapters and sections of large documents. Reusable content, such as warranty statements, hazard statements, and others may be stored once and included in multiple instances in final document assemblies. Individual words and phrases may be stored separately so that they may be updated as needed before final assemblies. Each of these situations requires a robust linking mechanism.

7.2.5 Authoring and workflow requirements

Organizations shall specify how an audit trail of changes and comments shall be maintained throughout the information lifecycle.

Organizations shall specify the elements of their information lifecycle management process, including importing legacy content, enabling authoring by both full-time professional authors and casual contributors, supporting simultaneous review by groups of reviewers and electronic signature of approvers, assembling content into larger deliverables, releasing approved content for publication, and archiving deliverables at end-of-life.

In preparing authoring and workflow requirements, organizations should identify how authors, reviewers, and approvers expect to work with the CCMS. The organizations may capture expectations as use cases or user stories.

8 Content management project plan

8.1 Implementation plan

The elements of the content management project shall be planned by the organization. The organization shall establish an implementation plan including the following activities:

- Define the information model
- Develop authoring guidelines
- Schedule activities, deliverables, and responsibilities
- Conduct training
- Develop stylesheets
- Conduct a pilot project
- Organizational rollout

A CCMS is not necessary for a successful planning phase of an implementation. A CCMS provides significant functionality, especially to support large volumes of content and translation management, but is not necessary for success.

8.2 Information model

An organization shall create an information model to specify the document types, information types, and content units that will be managed. An information model defines the structure of the content to be created and managed in the content management environment.

The information model shall include the definitions of the information items, information types, and content units to be created. In an XML authoring environment, the information model shall define the use of each XML element in the authoring environment.

Creating and approving an information model allows an organization to define the structure of the documents stored in the CCMS.

An information model specifies the structure of the content to be developed by the information-development organization. It may include references to specific published standards, such as OASIS DocBook, OASIS DITA, or S1000D, but may also be specifically designed for an organization's publications as a proprietary model.

The purpose of the information model is to promote consistency in the development of well-structured content objects that best meet the needs of the information users. By following a well-constructed information model, content developers throughout the organization are able to create uniform content that is capable of use in a wide variety of output types and among a wide variety of information products.

For organizations using a public standard, the information model describes the specific implementation of the public standard within the organization. In most cases, an organization may use a subset of the public standard or may add specializations or customizations of that standard to meet specific internal requirements.

8.3 Information model specification

An information model that supports structured authoring shall specify the following details about the organization's required content structures:

- definitions of each generic information item such as description, plan, policy, procedure, report, request, and specification and descriptions of the typical content and sequence of content;

NOTE: It may be useful to provide outlines and templates to assist in the development of each document type. ISO/IEC/IEEE 15289 includes recommended content for information items.

- definitions of the structures for collections of content such as chapters, parts, sections, books, embedded user assistance;
- definitions of each information type, such as overview, task, reference, and concept, explaining the purpose of each information type with reference to the appropriate audience;
- structure of each information type with required and optional content units and required and optional sequence of content units in the information type;
- definitions of each of the content units, such as sections, lists, paragraphs, tables, illustrations, examples, hazard statements, specifying how each is to be used to build each information type;
- definitions of elements, such as keyword, index term, citation, user input, and variable used within larger content units to specify words or phrases;
- definitions of terms to be used as metadata to describe each content unit, such as author, date created, subject matter, and version;
- mechanisms to be employed when reusing variables, such as words, phrases, or full content units across information or document types;
- file-naming conventions for both text and graphics to support consistency and retrieval.

These structures are illustrated in Tables 1 and 2.

Table 1 — Content data types in a CCMS information model

Data type	Purpose	Examples
Generic information item (archetype)	Provide outlines or lists of typical/required content	Description, plan, policy, procedure, report, request, specification
Collection of content	Organize content for use in an output document or document set	Chapter, part, section, book, help topic
Information type	Convey information for a specific purpose	Overview, task, reference, concept
Content unit	Build an information type with smaller components	Paragraph of text, table, list, illustration, example, hazard statement
Element	Specify words or phrases	Keyword, index term, citation, user input, variable

Table 2 —Templates for CCMS metadata

Template	Specifies	Examples
Information type	Required and optional sequence of required and optional content units	Task, concept, reference, troubleshooting, process, procedure, fact
Content unit	Metadata to retrieve the content unit	Author, date created/revised, subject

Template	Specifies	Examples
		matter, version
Variable insertion	Method for storing, retrieving, and using variables	Changing words, phrases, or content units in an information item or content unit
File name	Convention for storage and retrieval of files containing text and graphics	

In an XML authoring environment, the information model specifies the XML markup to be used for each content object from information items, content collections, content maps, content units with those topics and collections, and content units identifying specific words and phrases. The information model helps to apply the XML markup consistently to the organization's content so that content output can be generated using automated style sheets and that content objects can be searched and manipulated by authors using a content management database.

XML markup is supported through the use of an XML schema or XML Document Type Definition (DTD). A schema or DTD validates the structure of the document or topic. The DTD or schema can be used to identify invalid content. In some cases, the invalid content may still be allowed to be entered into the CMS but should be marked as invalid.

8.4 Authoring guidelines

Organizations shall develop authoring guidelines to provide instructional material to support the development of content that conforms to the information model. Guidance shall be provided on the correct selection of information types, the correct implementation of inline elements, the correct application of XML elements within each information type, and other requirements that help to specify the organization's XML standards implementation.

Authoring guidelines assist authors in planning and developing each document item, information type, and content unit required in the information-development lifecycle. Authoring guidelines instantiate the information model in a form that provides assistance and training for authors. The goal of authoring guidelines is to ensure that authors understand and conform to the structures defined in the information model. Authoring guidelines may also provide guidance for authors who are developing unstructured content. They may include guidance on proper use of terminology, writing style, grammar, and spelling, typical content of style guides.

Many organizations adopt authoring guidelines based on prominent published guidelines that emphasize producing quality technical information. In addition, such organizations supplement published guidelines with specific information dealing with the content the organization produces. However, traditional style guides are not sufficient to provide authoring guidelines for developing structured content.

8.4.1 Training authors

Authoring guidelines shall be used to train new authors in the rules established by the organization.

Authoring guidelines should include process guidelines for content management processes, such as creating variants and revisions, managing translation processes, and incorporating reuse mechanisms. Authoring guidelines should also include examples of developed content objects, including documents, topics, content units, and inline elements.

8.4.2 Code reviews

If organizations have implemented standards for XML-based authoring, they shall conduct code reviews of XML elements to support conformance with the information model. XML validation engines test conformance with the XML schema or Document Type Definition (DTD), but do not test the coding practices specified by the organization in its information model.

Code reviews should be conducted by individuals with expertise in the structured information model and the authoring guidelines instantiated in the organization. Code reviews may also be conducted using the Schematron standard, in which coding requirements are explicitly defined in the Schematron system.

Schematron is part of ISO/IEC 19757-3:2006, *Document Schema Definition Language - Part 3. Rule-based validation - Schematron.* Using Schematron, an organization can develop an additional rule set that tests the presence or absence of elements or sets of elements in the XML source. Schematron testing supplements, but does not replace, expert code reviews.

8.5 Reuse strategy

So that content is authored only once, organizations shall require that authors maintain records of content objects created for specific purposes or query the CCMS for content that matches a proposed new content object before that new object is created. Content objects that are intended for reuse may be stored in special "reuse" folders if these are available in the CCMS and should be labeled with reuse metadata.

The objective of content management is to create content once and use it, as needed, in multiple deliverables. Content can be reused either as complete modules or topics or as fragmentary content objects such as hazard statements, lists, paragraphs, words, and phrases.

Content reuse enables authors to update content in only one source, thereby maintaining the consistency and accuracy of the source content. Content reuse enables the organization to translate content only once from the source into the target languages, ensuring that the same content is translated consistently and accurately in each target language.

8.5.1 Content inclusion

Organizations shall identify and designate content objects that are frequently used in multiple deliverables and describe these content objects with metadata so that they are easily located for reuse.

Content inclusion mechanisms, which are often specific to the authoring platform or to the XML standard being used to author, permit content to be inserted in a target file from a separately maintained source file at time of publication. Consequently, the included content may be maintained in one source rather than repeatedly maintained as multiple targets. Maintenance of content objects in a single source reduces maintenance and translation costs and improves consistency, accuracy, and quality.

8.5.2 Content variables

Organizations shall develop policies to direct how authors use variables to allow substitution of words, phrases, or other content fragments into content objects.

Variable substitution mechanisms allow lists of terms or phrases to be maintained in separate source files and inserted into content objects at publication. Variable substitution permits the same content to be used in multiple deliverables that need small changes to content, such as product or system names or user interface terminology that may be changed frequently during the information-development life cycle. Variable substitution also may enable the same source content to be used for multiple deliverables that require only certain variables be inserted to correspond to a specific set of products, systems, or versions.

Policies should account for the possibility of mistranslation of sentences containing variables.

8.5.3 Content conditional processing

Organizations shall establish policies on the use of variables in source content objects that may be filtered at publication. Conditional processing uses metadata to identify alternative words, phrases, paragraphs, lists, graphics, or other components that are included in source content objects. At publication, the metadata is used to filter the content so that only one version of the source content is published, corresponding to the selected filter.

The presence of multiple variables in the source content objects enables authors to create one content object this is used for multiple outputs, functioning as a single source. The content objects containing multiple conditional variables may be updated once, reducing maintenance and translation costs and improving consistency, accuracy, and quality.

CCMSs may facilitate a variety of reuse mechanisms to assist authors in maintaining single sources of content and assist organizations in establishing reuse policies and practices. Organizations should maintain a balance between the usefulness and cost effectiveness of reuse mechanisms and risk of developing overly complex mechanisms that are difficult to maintain.

8.6 Metadata schema

Metadata are added to content so that the content can be searched and retrieved quickly and effectively. Metadata describe the content of the objects in terms that are meaningful to the users.

An organization shall establish a model that defines the metadata it will use to describe objects in the CCMS as well as the metadata to describe content delivered to customers.

A metadata schema shall be developed to label content components so that they may be effectively searched and retrieved and used to support automated processes for selecting or filtering content objects in final publications.

To create a metadata schema, an organization shall examine its content and identify the terms that best describe that content for one or more user communities. User communities may include:

- authors who need to search the CCMS for content related to their writing assignments;
- internal users who need to search the CCMS for content related to their work;
- consumers of content outside the organization who may use metadata to retrieve content best associated with their information needs.

When associating metadata with content, rather than storing it in the objects themselves, organizations should consider whether the metadata should be translated. Metadata that describe a category of content or type of task may be translated to facilitate searches by readers of that language. However, if a CCMS does not enable translation of user-specific metadata stored in the CCMS without the creation of additional, language-specific metadata fields, a process should be designed to export those fields for translation.

Metadata need not be applied uniformly to content within a CCMS. Rather, different metadata elements and/or values may be associated with different object types within the CCMS, such as library, publication, map, topic, graphic, template, or content unit. At the publication level, metadata may include information such as copyright dates, publisher information, volume, and part numbers. Topic metadata may include product name and version, brand, category, and keywords. Metadata associated with higher-level object types, such as a publication, should apply to the content within that object type. It is important that an organization understand how metadata values are transferred among nested objects in the publishing system they have adopted. Organizations should know if metadata values specified at a higher level add to or overwrite values specified within nested objects.

A metadata schema may be flat or hierarchical. In a flat structure, metadata is simply associated directly with the object type. In a hierarchical structure different metadata may be applied to different levels of the object. Certain metadata may apply to the base content, but other metadata may be associated only with a specific version, language, or workflow stage of the object.

EXAMPLE Version 1 of a topic includes information about a specific feature; if that feature is removed for version 2, metadata about that feature is also removed. If metadata is retained in version 1 of the topic, the version control system stores metadata associated with a specific version of a topic.

A metadata schema should define the approved values for each attribute. A list of possible values for each attribute reduces the risk of authors specifying similar, but different, values for the attribute making it difficult to employ a metadata-specific search. Each item in a list should be distinct and clearly defined so that authors know which values apply. If metadata values cannot be predefined, the information model should include examples of the form the values should take to be valid. A quality management system may be used to validate metadata values.

NOTE See ISO/IEC 11179-1:2004 Information technology — Metadata registries (MDR) — Part 1: Framework:

Metadata is designated as required or optional. A minimal set of required metadata should include administrative metadata and a basic set of descriptive metadata as determined by the information model.

8.6.1 Administrative metadata

Organizations shall specify the metadata that is collected and maintained by the CCMS.

Administrative metadata shall include the following:

- name of each individual who has contributed content or interacted with content in the CCMS, including authors, editors, reviewers, and approvers;
- dates and times associated with every action taken on the content in the CCMS, including the date and time of every change committed to the CCMS for a component;
- a field for optional comments from individuals contributing or interacting with content to explain the reasons for the changes made to a component.

Organizations should understand that some metadata is automatically collected and maintained by the CCMS. Administrative metadata are typically maintained through the life of the components in the CCMS. This metadata should be available if the content is moved to a new CCMS without loss of information.

8.6.2 Descriptive metadata

Authors and others responsible for identifying the attributes and values associated with the content shall be required to apply metadata to content objects. Such application of metadata may be supported by automated metadata systems. Metadata pairs of attributes and values may include values to label the subject areas of the content. Examples of metadata values added by authors to the content are as follows:

- Product attribute with values consisting of product names
- Audience attribute with values consisting of different audience types
- Component attribute with values consisting of hardware components
- Platform attribute with values consisting of the platforms supported by the products
- Other attribute and value pairs that are needed to describe the content in both documents and topics

In developing a metadata model, organizations should aim to optimize subsequent searching in published content by including metadata that facilitate searches. Metadata associated with documents and topics may be exported with that content for delivery to users. The effectiveness of search systems that use metadata to focus end-user search and retrieval is dependent on the quality and consistency of the metadata and its regular addition to content objects.

Descriptive metadata is likely to change as products are updated, user communities realign, and standard terminology changes. The organization should review its metadata schema periodically and update.

NOTE Standards are available that provide guidance on controlled vocabularies and thesauri for specific subject areas. ISO 25964-1:201 Information and documentation — Thesauri and interoperability with other vocabularies — Part 1: Thesauri for information retrieval provides guidance for the development of thesauri controlled vocabularies.

8.6.3 Processing metadata

Authors and others responsible for identifying the attributes and values associated with publishing output may be required to apply metadata to objects, which may include attributes with values for:

- layout specifications or styles;
- rules for variables in content;
- location and inclusion of external data.

8.7 Workflow specification

The organization shall specify a workflow. A workflow specification includes the process of setting up and using a workflow, including the automation of that workflow in a CCMS.

The workflow specification defines each action that occurs during the life cycle from the initiation of a project through the release of content to the consumer, sustaining the content in a reuse environment, and archiving the content for permanent storage. Once specified, the workflow may be partially automated using the workflow functionality of the CCMS.

The goal of the information-development life cycle is to ensure that the quality of the content developed is maintained and that work occurs in a timely manner, often expressed as conformance to required deadlines.

8.7.1 Workflow approvals

The workflow system shall allow for repeating the writing/editing cycle. The organization shall establish a policy that content is not released to the users until required approvals are completed.

Typical workflows establish routing for the following activities:

- Project initiation
- Content editing and proofreading
- Content technical review
- Content testing
- Content approval
- Content translation
- Content publication
- Content sustainment and reuse
- Content archiving after initial and subsequent releases

If the workflow specification is automated, the workflow system or multiple workflow systems enable each action to be initiated, with assigned deadlines if applicable, and automatically notifying those responsible for each stage of the process that input is required.

The workflow system enables project managers and others participating in the workflow to track the status of each action, monitoring its progress and completion, including sending reminders when the actions have not been completed in a timely manner.

Projects may be initiated in the CCMS by assigning the development of content, including topics or complete documents, to the appropriate authors. Once assigned, the workflow system notifies the authors of their assignments and provides access to the CCMS in which those assignments will be completed. The assignments may specify the deadlines for one or more drafts of the assigned content.

When authors have completed their assignments, they use the workflow system to automatically forward the content to those responsible for editing or proofreading. The workflow system notifies the assigned editors or proofreaders of their assignments and tracks their completion. After editorial tasks are complete, the content may be returned to the authors for resolution of changes.

The same process occurs for subsequent steps in the information-development life cycle, including reviews by technical experts, work by test professionals to verify and validate the accuracy of content, and approvals by those responsible.

If the organization requires that a pool of editors, proofreaders, technical reviewers, testers, or approvers be available, the workflow system may be designed to accommodate the specification of a pool so that one member of the pool may accept the assignment, thereby removing the action from the project work of the additional members of the pool.

8.7.2 Translation workflow

If translation of the content is required by the process, the project manager or other responsible party shall initiate the development of a translation package by the CCMS. A translation package may be created following the OASIS XLIFF specification. The CCMS creates a translation package that shall be made available to the organization responsible for translation and accepts the returned package when the translation is complete.

8.7.3 Workflow completion

Finally, the workflow system supports the final steps in the workflow specification, including publication of the content in one or multiple formats and archiving the published content for future reference.

Although workflow systems as part of a CCMS support the automation of the workflow specification, project managers should take care to avoid overly complex workflow processes. Beginning with a simple workflow process and automating it initially will produce better results than developing a complex workflow that is difficult to automate.

8.8 Schedule of activities, deliverables, and responsibilities

The organization shall develop a schedule of activities to be conducted, the deliverables to be produced, and the responsibilities of team members in the implementation plan.

8.9 Training plan

The organization shall develop a plan for the training required for the participants in the pilot project and through an organizational rollout of the content management environment. Training shall include the information model as well as the systems used for authoring, publishing, and managing content objects.

The organization should schedule the required training immediately preceding the time when it will be used in the implementation schedule. Training provided too early often needs to be repeated to avoid risking the project implementation plan's success.

Many methods of training can be used concurrently, including a buddy system, formal training, informal FAQ sessions, and so on. Each method meets a different need and together keeps everyone working effectively.

8.10 Stylesheet development

The organization shall develop stylesheets to support the formatting and publication of approved content. In a structured authoring environment, each deliverable produced has a style standard that specifies the page layout, font selection, and other aspects of well-defined final deliverables. In an XML structured authoring environment, the final design and layout of deliverables are established using coded stylesheets, often using the XSL-FO standard for PDF (Portable Document Format) publications and the CSS (Cascading Style Sheet) for HTML publications. Individual authors are unable to affect the final styles of the documents; those styles are determined by the organization’s standard stylesheets.

8.11 Pilot project specification

Organizations shall specify one or more pilot projects to validate and verify the development activities before they are introduced to the larger organization. The selection of the pilot project should be based upon these criteria:

- exercises the information model;
- is important to the organization, but not critical;
- can be completed in three to four months;
- demonstrates the capabilities of the CCMS to improve productivity, decrease costs, and improve quality;
- provides an opportunity to train staff members who will lead subsequent projects in an organization-wide rollout.

In addition, criteria shall be established to evaluate the pilot project. Criteria may include the ability to implement the information model successfully and the ability to apply the selected tools successfully.

8.12 Organizational rollout

After the pilot project is complete, the organization may extend the implementation plan to include the next set of projects for a rollout through the organization. Staff members trained in the pilot project may be selected as project leads for the additional projects to be included in the next stages of project implementation, while staff continue to be trained and new tools introduced.

9 Information development

9.1 Content conversion

Information development addresses the conversion of legacy content and the development through authoring of content structured according to the rules established in the information model.

Organizations shall provide policies and procedures for converting unstructured content to structured components.

Most organizations that inaugurate a content management project have unstructured content, content that does not conform to a standard XML schema or Document Type Definition (DTD). In addition, much content is poorly organized, in that it is written inconsistently and does not conform to standards, even those that have been defined by the organization.

Organizations that prepare to implement a structured content development process shall perform a content inventory to identify the following:

- unstructured content to be converted in conformance with a structured information model;
- content to be maintained without conversion, such as content nearing end-of-life;
- content to be rewritten to conform to a structured information model.

Well-organized existing content may be automatically converted to structured components that conform to an XML-based information model. So that the conversion process is successful, an organization may decide to better organize existing content in the unstructured authoring environment before conversion or may decide to repair structural problems following conversion. Automated conversion works best when existing content is well organized and follows authoring guidelines.

Poorly organized existing content should be rewritten to conform to the information model before conversion. After content is converted to a structured authoring environment, authors find it more difficult to restructure the new components than if they revised content in their familiar authoring environment. However, unstructured authoring environments provide no mechanisms to enforce and validate structure. Therefore, rewriting content that should be strictly structured in an unstructured authoring environment may be less successful than desired.

Some existing content in its legacy unstructured form, especially content that is near its end-of-life, should not be converted to a structured form at all. Content that is unlikely to be updated or have only minor changes may best remain in its existing form.

9.2 Content authoring

Mechanisms for authoring content in a component content management environment may include both mechanisms to support word processing and desktop publishing, as well as XML-based structured authoring.

Policies for content authoring in a content management environment may permit structured or unstructured content to be included. Structured authoring prescribes the sequence and content required for specific document and topic information types. Structured authoring can be enforced by validated XML schema or Document Type Definitions (DTDs). Unstructured authoring does not prescribe structure but allows unstructured content to be included in the CCMS. Many component content management environments include both structured and unstructured content.

9.2.1 Structured authoring

If structured authoring is used, organizations shall require that components conform to structures defined in the organization's information model. The organization shall define the purpose of each information type and content unit and its required and optional subject areas.

An organization implementing structured authoring shall require that authors label content elements using semantic labeling and following the specification provided in an XML schema or a DTD and manifested in the organization's information model.

In structured content, components are labeled so that they can be programmatically manipulated to create a variety of outputs. The output is not restricted to the sequence in which the content was created by the author. Semantically labeled content elements may be addressed and restructured to correspond to the needs of the organization or the users.

Structured authoring is supported by XML editing tools. Authors conform to the requirements of the XML schema or DTD through the validation routines of the XML editor. Templates designed to guide authors to produce well-formed and valid XML that also conforms to the requirements of the organization’s information model may also be used to support XML-based authoring.

9.2.2 Unstructured authoring

If unstructured authoring is used, organizations shall define the content types of unstructured content that may be authored and managed by the CCMS. The organization may also define informal structures for certain content types.

By definition, unstructured authoring is uncontrolled by an underlying well-defined formal architecture. Rather, the content is organized according to the decisions of individual authors. Authors of unstructured content may indeed follow guidelines, but the resulting topics are considered unstructured because the individual content elements cannot be individually addressed because they are not labeled.

Word processing and desktop publishing systems produce unstructured content that has low semantic value, low consistency, and low automation potential. Content structured in such systems is imitated through formatting controls, enabling authors to produce consistent informal structures if they follow the rules for formatting. Content defined by heading levels, lists, and paragraphs can be programmatically addressed and manipulated in limited fashion to produce a variety of outputs.

9.2.3 Content granularity

The organization should define the optimal length of topics, depending upon the needs of its audience and the requirements of its content reuse strategy. Optimal lengths may be associated with specific content types.

Authors often ask what constitutes an appropriate length for an individual topic. The general rule is that a topic answers a single question. A concept defines what something is. A task explains how to complete a procedure and reach a goal. A reference topic provides facts that the user needs to successfully complete a task or make a decision. Additional information types, such as troubleshooting, may be used to answer a specific user question or assist a user in reaching a goal.

A user can need a complete set of information to answer a single question, which may be found in a primary topic and its subsections. On the other hand, a user can need a topic to be brief, able to be comprehended in a few minutes. A more complex piece of information may be contained in multiple topics. The length of a topic requires a balance between these two constraints: quick reference and thoroughness.

The length of a topic may also be influenced by the experience and sophistication of the user. A more experienced and sophisticated user may be more interested in and tolerant of the details of a more complex topic than a less experienced beginning user.

Providing users with links among topics is a mechanism to increase the coverage of a topic while at the same time keeping individual topics at a minimal length. However, linking mechanisms, such as cross references, should be minimized to avoid sending the user in multiple directions to answer a single question or solve a single problem.

Topic size is also a factor of the reuse environment implemented by the organization. A brief topic is more amenable to reuse than a lengthy and complex topic. Small topics may be joined together to provide a longer discussion of a subject for a particular user community. Small topics may remain separate to accommodate those needing only a subset of the information.

Topic size may be governed by the need to assign unique metadata classes for precise retrieval of topics. However, topic size is governed by the concept that a topic answers one reader question, such as “what is …?”, “how do I …?”, and so on.

10 Management and control

10.1 Managing quality

Quality shall be monitored through the information development life cycle as defined by the workflow specification.

NOTE ISO/IEC/IEEE 26513:2009, Systems and software engineering – Requirements for testers and assessors of user documentation, has a detailed specification of review and assessment activities.

The organization shall establish guidelines and minimal requirements for quality review during the following process steps:

- Project initiation
- Content development
- Content editing and proofreading
- Content code review
- Technical review
- Content validation and verification

- Translation review
- Publication review

In each step of the process, the organization should define standards of quality, such as the following:

- process requirements:
  - Quality requirements are defined for each step in the content management workflow.
  - Steps in the process are done according to the content management workflow.
  - Steps in the process are completed according to the quality requirements defined for that process.
- information model, authoring guidelines, style guide, and terminology consistency:
  - Components conform to the structural and XML coding requirements of the information model.
  - Components conform to the authoring guidelines for the correct application of organizational standards.
  - Components conform to the appropriate style guides required by the organization.
  - Components conform to the requirements for the consistent use of terminology as established by the organization.
- technical review, validation, and verification:
  - Components were reviewed by designated subject-matter experts (SME) to confirm the accuracy of the content and its conformance to system or product specifications.
  - Components have been validated and verified as technically accurate by a process in which procedural information is tested with the system or product that is delivered to the consumer.
  - Components that contain sample code or other verifiable content have been validated and verified as technically accurate.
- translation review, validation, and verification
  - Components that have been translated into target languages from the original source language have been reviewed and verified by qualified in-country SME.
  - Terminology used in the source language has been translated into the target languages based on the content of a verified organizational terminology database.
  - A terminology database consisting of pairs of source and target language terminology has been verified by in-country SME.
- publication review
  - All components are validated for adherence to the organizational requirements for final publication review before being released to the consumers.

10.2 Review and approval of content

A CCMS may be used to automate aspects of the quality management process.

Adherence to the consistency requirements of an information model, authoring guidelines, a style guide, and terminology standards may be automated using quality management systems linked to the authoring environment, including XML editing. Such systems should be configured to conform to the organization's standards and guidelines in order to assist authors in maintaining content quality. However, the use of quality management systems is often incomplete, and documents or topics may need a human review.

Other automation in the CCMS may be limited to the notification of parties associated with actions in the workflow specification. However, review activities may be supported by review systems that facilitate reviews by SME within the CCMS. Such reviews may facilitate simultaneous reviews by several experts, thereby

reducing the review time and complexity. Simultaneous reviews permit experts to view each other's comments as they are made.

Review automation should facilitate storing review comments and their resolution within the CCMS and automatically create reports of the review actions and resolutions that may be required by regulatory bodies.

A CCMS may facilitate formal approvals of documents or topics through the use of electronic signatures.

The workflow specification may include the notification of an organization's official approver or a pool of approvers for a document or topic. If a pool of approvers is available, the first available individual may accept the approval assignment, thereby blocking others in the pool from duplicating the approval process. If multiple approvals are required, each individual in the pool of approvers shall be notified that the document or topic is ready for approval. In this case, the approval process is not complete until required approvers have completed their reviews and designated their approval.

A CCMS may store electronic signatures of each designated approver within the workflow specification. The electronic signatures are permanently associated with the final approved version of the document or topic within the CCMS.

10.3 Search and retrieval

Organizations shall define how content authors search for components in the CCMS, including full-text search, metadata search, XML element search, and combinations of these methods. Organizations shall provide guidance for authors and others searching for content on the search mechanisms available and how to use them.

Organizations shall also develop policies and practices to assist authors to search the CCMS for existing topics that they may use in new publications as they plan new information-development projects. Maintaining an inventory of topics that address particular subject matter areas as part of the planning process may reduce the practice of searching for topics at random. Authors are more likely to plan effectively from well-structured topic inventories rather than through random searches.

10.4 Localization and translation

Organizations shall indicate which components are to be reused so that the components are updated once and translated once.

CCMSs support the efficient synchronization and management of source and target language XML components. Changes in source XML language components result in status changes that reflect the need for translation updates of target language components. The synchronization of source and target components occurs through the link-management function of the CCMS. In component content management, every component is also linked to maintain synchronization between source and target languages.

Managing localization practices in a CCMS enables organizations to:

- write and translate content once using content reuse mechanisms;
- use graphic formats that allow text to be stored independently of graphic content. In the case of vector graphics, an example format is Scalable Vector Graphics (SVG).

NOTE The SVG specification is an open standard developed by the World Wide Web Consortium (W3C).

- using CCMS functions, isolate components that have changed from those with no changes and include only the changed components in translation jobs.

XML-based reuse mechanisms allow organizations to establish policies that require authors to isolate and maintain collections of reusable components that are referenced by XML topics in both source and target languages.

10.4.1 Content management for translation

Organizations that deliver translated content shall establish workflow policies and practices to prepare content for translation:

- Allow localization specialists or those responsible for the translation process to be assigned to the content-development workflow so that they are notified when content is ready for translation.
- Allow localization specialists to prepare translation packages using the functions of the CCMS.
- Send the translation packages to the appropriate localization system providers.
- Receive the translation packages back from the localization system providers and import the translated components to the CCMS.
- Include review and validation of translated content in the content-development workflow, if required.

10.4.2 Publication of translated content

Organizations that deliver translated content shall establish policies and practices to publish content in target languages. An organization may:

- invoke the publishing pipeline in the CCMS to produce appropriate output in selected target languages.
- export the translated content from the CCMS to be published locally at the desktop or through a separate publishing system.

10.4.3 Translation of vector graphics

If vector graphics are stored in the CCMS, they may be converted to Scalable Vector Graphics (SVG), an XML format, from which text may be automatically extracted for translation. Translated text is then added back into the graphic files, producing final graphics that may be incorporated into translated documents. Text is extracted only if the source graphic is in a format that supports layers and the text is in a separate layer.

10.5 Content deletion

The organization shall establish policies and practices for deleting components from the CCMS. Component deletion shall be reserved for authorized administrative individuals to avoid deleting components inappropriately.

CCMSs incorporate a function that automatically checks where a component is used in the CCMS. Given the nature of XML-based component authoring for reuse, in which components are used in more than one publication or delivery environment, an administrator shall review the "Where used?" results so that deleting a specific component does not indiscriminately delete a component that is being used in multiple publications. If a component is used in multiple publications, the owners of those publications shall be consulted before the component is deleted. A deletion activity shall request confirmation that the translated versions of the component no longer need to be stored in the CCMS.

In general, given the low cost of storage, components need never be deleted from a CCMS. However, at times careful cleanup of the CCMS can be needed. That cleanup function should be conducted with a complete understanding of how components are stored and how links are maintained among components. Some CCMSs will not allow a component to be deleted if it is referenced by other components in the CCMS or if it is used in more than one publication. Such systems may be configured so that authors are not able to delete components but may flag components for deletion after careful research by the CCMS administrator.

Because CCMSs maintain version control, they store multiple historical versions of every component. Deleting the most recent component deletes the version record and the previous versions of that component. It is likely that earlier versions of the component are used in earlier versions of published output from the CCMS.

Therefore, it is always unwise to delete topics and the previous versions until the components have been archived.

CCMSs link source and target versions of a component. If the source component is deleted, the CCMS also deletes the translations of that component.

Despite the concerns discussed with component deletion, a CCMS may become overwhelmed with components that are no longer used. An option is to place these components in an archive area of the CCMS or to formally archive them to another system.

10.6 Content and component archiving

The organization shall define policies and practices for archiving digital content.

CCMSs manage archiving of electronic components depending on the requirements of the organization. Some organizations require that components be archived outside the CCMS, typically in another CCMS designed specifically for archiving purposes. Some organizations require that an archived copy of a released document or set of components be managed within the CCMS so that a copy of record is maintained.

Due to the low cost of storage, component archiving does not ordinarily occur simply to save space in the CCMS.

If component archiving is necessary, organizations shall maintain policies that explain why and when components should be archived. In a CCMS, the components associated with a release, including graphics, multiple media, and referenced components in the source and the target languages shall be maintained through archiving, so that the release can be recreated in the future.

If components are to be archived to a CMS developed solely for archiving, organizations should consider its accessibility, format, regulatory policies, and other issues in determining their policies. See Digital Archiving policy in ISO 14721:2003. Archive policies may include:

- requirements that specify when components are to be archived, such as component age, component size, components about products or systems no longer supported, components about product or system versions no longer supported, components about products or systems no longer owned by the organization;
- format in which the components are to be stored;
- how the archived components are to be accessed;
- regulatory or legal requirements affecting the time that components shall be maintained in an archive CMS.

Components may also be archived internally in the CCMS so that a copy of record of a publication release is preserved.

11 Publication

11.1 Release management

The organization shall define the process for publishing source content using the publishing capabilities of the CCMS.

Organizations that support multiple and simultaneous product versions shall establish policies and practices to manage multiple versions of components in the CCMS.

Many organizations support multiple product release versions simultaneously. Authors simultaneously update and edit content that is used across multiple versions of the product or system documentation and may merge those versions at a future time.

Branching and merging is widely used outside the software industry as well. Branching and merging can also be used to manage variants as well.

To support multiple versions of a product in the documentation, components are branched to account for ongoing changes and then merged back into a main topic when the next product release occurs. When this process is handled manually, often by maintaining multiple versions of a component, merging is error-prone, tedious, and time-consuming, as the authors examine multiple copies of the component and discover that some variants are incorporated and others are not. This manual process may be automated to the extent that changes among branched components do not conflict. If changes conflict, manual intervention is necessary.

11.2 Version management

Organizations shall maintain policies and practices to avoid conflicts among authors of concurrent versions of components. These policies and practices shall include:

- under what circumstances to create and manage branches (multiple versions of the same base components);
- who is responsible for creating branches of components;
- how to create and label the component branches in the CCMS.

The organization shall establish a policy and process for merging branches in accordance with the functionality of the CCMS. Once the branches are created, organizations shall establish practices for authors to follow in updating and editing branched components. Each branch is a unique instance of the components that may be independently edited. Authors working on the set of branched components work collaboratively, ensuring that decisions on potentially shared components are made consistently.

At some point, the organization may decide to merge branches into a new single version of the component. The organization may also establish a policy that some or every branch of components be maintained or discontinued.

Branched components may be merged automatically, which means that the CCMS functions to compare the branched components and institute automatically the non-conflicting changes. A non-conflicting change is typically unique, without multiple changes to the specific text.

When conflicts in branched components or other areas are discovered, the organization should have procedures and a designated authority, such as the content owner, a change control board, or by informal collaboration to decide which changes should be implemented and which should be resolved through collaboration.

11.3 Publication of content

The organization shall define the output formats (that is PDF, HTML, user assistance) required by their users and establish a set of publishing pipelines to enable authors to select and implement a suite of publication types through:

- functionality of the CCMS;
- word processing and desktop publishing applications;
- external processing systems;
- mixture of these methods.

For documents, the publishing of output is handled within the application that was used to create the source documents. Word processing and desktop publishing files may be printed directly or transposed to PDF using the application functions themselves. For XML files, the publishing of output may also be handled within the application or may be handled through an external processing system or through the CCMS.

Documents that originate in standalone applications for word processing or desktop publishing are processed for publication at the desktop. Some organizations choose to publish XML content at the desktop as well, which may introduce inefficiencies and the inability to scale to large volumes of content. Desktop publishing may also introduce performance problems because of file size. File management may also become complex because of requirements for local versions of publishing software and templates and stylesheets.

To avoid duplication of effort and technology, the organization may establish a centralized publishing responsibility. The centralized publishing authority may elect to publish through an external processing system or through the publishing functionality of the CCMS.

Organizations that develop XML topic-based content may publish individually on the desktop using the XML editor applications, may publish through external dedicated processing systems, or may publish through functionality integrated into the CCMS. External dedicated processing systems provide performance advantages because they are typically housed in independent server environments that exceed desktop publishing capacities. CCMSs provide the same capacity without removing the source material from the CCMS. They also manage maintenance of the output files in the CCMS as an archive of the published version. The source files that remain in the CCMS allow the published version to be generated on demand. Because the publishing software, templates, and stylesheets are stored in the CCMS, the organization can maintain the publishing standards for every publication.

CCMSs support the publication of content for distribution outside of the working content, typically to a corporate website, to print, to embedded user assistance, or to multiple mobile devices. Dynamic links may be established between the CCMS and the websites or other delivery mechanisms so that content may be automatically updated or may allow the users to select content for a custom deliverable that is then generated automatically from the most recent approved versions stored in the CCMS.

12 Component Content Management System requirements

12.1 General

To manage content efficiently, provide distributed and controlled access, and maintain a single source of authoritative content, components developed by an organization shall be stored in a CCMS. XML structured content shall be managed in a database that stores and is capable of addressing individual components for identification and retrieval. The input, storage, removal, and output of the components in a CCMS shall be controlled through automated functions.

Using a CCMS allows each item of stored content to be independently maintained so that it may be accessed and used by individuals with the proper permissions. Each component may be added to the CCMS after authoring or may originate in the CCMS.

Database support for a CCMS may be of several types:

- Relational database
- XML database
- Mixed database types that include both relational and XML structures
- Object-oriented databases

The type of database selected depends on the needs of the organization.

Given the requirement that components be available for reuse and linking, a CCMS should manage components created using an XML structure, supported by a Document Type Definition (DTD) or a schema.

These systems are generally referred to as CCMSs because they are able to address and manage the individual components that are used to build larger content objects. A content object may consist of paragraphs, lists, and tables as well as more detailed semantically named structures such as <steps> or <notes>, in conformance with the associated DTD or schema. A CCMS that can deconstruct large content objects into their components allows these components to be addressed and managed individually.

12.2 Component Content Management System framework

Components, content types, metadata structures, and versioning information in a CCMS that uses an underlying database have specific structural requirements stated in this clause.

12.2.1 General storage requirements

Regardless of the underlying database used, the CCMS shall store content as components in the CCMS. Components shall

- have an associated name (human-readable label) by which a component may be referenced;
- have a unique ID that shall be used to reference the component irrespective of its name or location in the CCMS;
- have an assigned Content Type;
- support setting and retrieving associated metadata.

When storing components, the CCMS shall provide the ability to retrieve components such that their content or linking relationships are not unexpectedly modified. These unexpected modifications include adding vendor specific information that may affect the ability of other systems to validate, transform, or integrate the content. For XML systems, adding processing instructions is not considered an unexpected modification.

12.2.2 Content types

The CCMS shall provide for the concept of a Content Type. A Content Type is a re-usable definition of settings for storage, metadata, workflow, and behavior. Content Types enable settings to be administered to components in a centralized and re-usable manner. A component shall only be associated with a single Content Type.

Content Types may support inheriting settings from other Content Types. The CCMS shall associate components with their given Content Type. Common reliable associations are provisioned from a basic component property (file extension, mime-type, or DTD definition).

12.2.3 Metadata structures

The CCMS shall support storing and retrieving metadata associated with components as "metadata fields." The CCMS shall support arbitrary textual content for metadata values. However, these values may be further controlled by applying constraints or typing to fields. A metadata structure, typing, or control mechanism that is not defined by the CCMS shall be administered through its associated Content Type.

Metadata fields should be associated via a key value pair.

The CCMS shall accept at least two basic forms of metadata: descriptive and administrative.

12.2.3.1 Administrative metadata

Administrative metadata is meta information which is provided to aid in managing components. The CCMS should provide the following administrative metadata values:

- Created date and time

- Last modified date and time
- Content type

CCMSs may provide additional administrative metadata, including:

- Owner user and group
- Permissions
- Component lifecycle state
- Validation status
- Rights management metadata

This metadata should be available if the content is moved to a new CCMS without loss of information.

12.2.3.2 Descriptive metadata

Descriptive metadata is meta information to aid in search, discovery, classification, and identification. Common descriptive metadata includes the following:

- Keywords
- Tags or labels
- Classification
- Other taxonomic values

The CCMS should support typed metadata values. Typing metadata values allows each metadata field to be assigned a specific information type. Using information typing of metadata values means that input is validated or provided from a controlled classification mechanism.

Taxonomic metadata definitions should support storing and retrieving hierarchical values.

12.2.3.3 Additional metadata requirements

The CCMS shall allow the metadata schema to be updated programmatically.

If some values cannot be directly modified by the user, the CCMS should treat them as "read only."

The CCMS may also support the ability to assign metadata to components based on the associated Content Type definition.

12.2.4 Organizational structures

The CCMS shall provide at least one primary mechanism to organize content into logical structures. The logical structure may be accomplished through folders, metadata, publication structure, or other means.

12.3 Component Content Management System management

CCMS management includes the following activities:

- Component creation and modification
- Import/export

- Archiving

12.3.1 Component creation and modification

The CCMS shall provide basic functions for managing content in the CCMS: Create, Read, Update, Delete, Rename, Move, and Copy.

Create allows a new component to be created in the CCMS based on a defined Content-Type. The component shall be created based on a human-readable name and the CCMS shall generate necessary administrative metadata. The CCMS shall also check that the component is valid with respect to the Content-Type definition. This may be accomplished via querying the user for needed information or automatically populating content and metadata based on predefined templates.

Read allows the content and metadata for a component to be retrieved for viewing by the user.

Update allows an existing components content or metadata to be updated. The CCMS shall validate the updated content and notify the user if problems are present. If validation fails, the CCMS may choose to deny the update. In this situation, the CCMS shall roll back changes that may have occurred during the update process.

Delete allows a component to be removed from the CCMS. Links to that component should result in a component-not-found exception. However, the CCMS shall retain that object's metadata and version information so that it may be restored to its previous state at a future time.

Rename allows a component's name to be changed. When a rename occurs, the CCMS shall update links to that component from other components as part of the process.

Move allows a component's apparent location in the primary organizational structure to be changed. When a move occurs, the CCMS shall update links to that component from other components as well as links from that component to other components, to reflect its new location in the CCMS.

When applying these functions, the CCMS shall update links to that component from other components as well as links from that component to other components during the process.

12.3.2 Import/export

The CCMS shall provide the ability to import components from and export components to a local file system. During the import process, components that do not exist in the CCMS shall be handled by the Create function. Components that already exist shall be handled by the Update function. In the event that an issue occurs during the import process, it is recommended that changes made during the process be rolled back.

The CCMS should also preserve the organizational structure of imported components.

12.3.2.1 Bulk export

To support bulk export, the CCMS shall provide a mechanism to download a set of files as a single package. When the user selects a folder or a set of files to export, the CCMS shall package and deliver those files to the user in a format that may be unpacked and used on a local file system. The CCMS shall support bulk export.

12.3.2.2 Dependency export

The CCMS shall allow dependency export.

The CCMS shall provide a mechanism for a user to download a packaged set of files and its dependencies based on internal linking structures.

The CCMS shall not export non-local dependencies or dependencies that are marked as 'external' in the linking structures. Links to websites, reference materials, or other content that does not exist locally shall not be exported.

12.3.2.3 Archiving

The CCMS shall create an archive or snapshot of a set of content and its subsequent dependencies at a specific time. An archive shall include source content, translated content, and published versions. Content or metadata in an archive shall not be modified. If users need to modify content, they shall do so by exporting or branching content from the archive.

Components in an archive, including content and metadata, shall be retrievable in the exact state in which they existed at the point of archival.

12.4 Content object management

Content object management encompasses requirements associated with check out/check in, link management, search, and versioning.

12.4.1 Check-out/check-in

The CCMS shall check components in and out of the CCMS. When a component is checked out, it shall be locked. For locked components, the CCMS shall indicate that the component is currently checked out by a specific user. The CCMS shall prevent other users from checking out a component that has been checked out by another user. If another user attempts to check out a locked component, the CCMS shall deny this operation and indicate the user who currently holds the lock.

When components are checked back in, the CCMS shall remove locks put on these components.

The CCMS shall allow single component check-out/check-in, as well as bulk check-out/check-in.

The CCMS shall allow administrators to unlock components regardless of which user originally locked them.

In the event of a CCMS restart, the CCMS shall retain locking information for components.

On check-in, the CCMS shall trigger an incremental version of the components either automatically or manually.

The CCMS shall connect authoring tools directly to the CCMS, without the user needing to checkout and download components manually, and shall have the ability to set components' check-out/check-in state.

The CCMS shall prevent users from moving, renaming, or performing other operations that may affect links contained in components that are currently checked out by other users.

EXAMPLE: If document A links to document B and document A is checked out, document B may not be renamed or moved until document A is checked back in.

12.4.2 Bulk check-out/check-in

The CCMS shall select and check out or check in an arbitrary set of components from the CCMS. The CCMS shall also support dependency-based check-out/check-in. When dependency-based check-out/check-in is invoked, the CCMS shall check-out/check-in the selected components, as well as components that are direct or indirect dependencies of those components.

The CCMS shall not allow a user to perform a dependency-based check-out/check-in if a component in the set of dependencies is checked out by a different user.

12.4.3 Link management

The CCMS shall manage links between components.

The link management system shall provide the following reporting capabilities:

- For each link, the system shall report to the user the type of link: relative, absolute, or external. For relative and absolute links, the system shall report if the link is broken. A link is considered broken if the CCMS cannot resolve the component being linked to.

- For each component, the system shall list links contained in that component and the destination component of each link.

- For each component the system shall list every link that references that component. This function is often called 'where-used.'

- For each component, the system shall list the components that are dependencies of that component.

The link management system shall automatically correct links referencing components when a component or set of components is renamed.

12.4.4 Search

The CCMS shall allow components to be found by querying the CCMS. When the CCMS is queried, the CCMS shall take the access control settings of the CCMS into consideration, filtering components from the result list to which the querying user may not have access.

In addition, the CCMS shall allow queries of the entire CCMS, as well as confining the query to a specific set of components. The CCMS shall allow the following methods of confining a search query to a:

- set of selected components and confined only to those components;

- single component and only to that component;

- single component as well as that component's dependencies and only the component and its dependencies.

When searching, matched results shall be scored based on their relevance to the search query. Higher scored results shall appear higher in the list of returned search matches.

The CCMS may also combine two or more querying methods to further refine the matching results.

12.4.4.1 Full text search

In a full text search query, the CCMS shall return matches based on text content in components.

The CCMS shall allow for entering more than one search term. The CCMS may consider phrases to be indivisible, as a single unit, if the user designates that the whole phrase should be used as an exact match.

The CCMS may also support advanced matching capabilities. Advanced full text matching may include:

- Near search
  - Matches are scored higher the closer together the terms appear in the component.
- Wildcard search
  - The user may specify a wildcard value in a search term that will match a set of characters that results in a match for that term.

EXAMPLE "digg*", would match "digged", "digging", "diggings", "digger", and "diggers".

- 'Regex' Regular expression (Regex) search
  - The search query is a regular expression.

- Phrase search
  - Treats the group of full text search terms as an ordered set that is to be matched to content in the source components in the same order as supplied in the query.
- Boolean search
  - For each term, the user may specify whether terms are a connected set with Boolean logic. This logic states if terms shall appear, may appear, or may not appear.

EXAMPLE This query states: Return documents that match terms “dog” and “cat” or match “horse” but do not contain “donkey”.

“(dog AND cat) OR (horse AND NOT donkey”).

The CCMS should report on which terms matched a given component and show a highlighted excerpt of the content where the match was found.

Full-text search shall include auto completion of search queries and spelling corrections or suggestions.

12.4.4.2 Full text search in XML repositories

For CCMSs supporting XML content, the CCMS shall:

- not consider XML tag names, namespace declarations, comments, processing instructions, or attribute names as part of the full text content being searched;
- tokenize component content on whitespace based on XML whitespace processing rules;
- tokenize component content based on the XML structure. Thus, the CCMS shall recognize that the tags between texts imply a break in the text token.

EXAMPLE In the following markup, the CCMS recognizes that “Item one” and “Item two” are independent clauses, even though they are not separated by whitespace or punctuation.

Item one
Item two

12.4.4.3 Full text search configuration

The CCMS should support search configuration for full text search on structured content. In a CCMS with XML content, the structure should be used to improve searching capabilities, such as.

- **Boosting matches**

  The CCMS should boost the score of matches found in particular parts of a component.

EXAMPLE: The CCMS is configured so that matches in the ‘title’ of the component are scored higher (given a higher ranking in search results) than those found in other parts of the document.

- **Ignoring matches**

  The CCMS should ignore matches found in designated parts of a component.

EXAMPLE: The CCMS may be configured to ignore matches found in alt text for images or information in files used for processing.

- **Rules for mixed content**

  Mixed content occurs in structured documents when markup splits words or phrases that would otherwise be considered contiguous. The CCMS should configure which elements split content that should be considered contiguous.

EXAMPLE: In the following structured content, the word “unclear” is split but should be considered a single entity.

This is unclear

12.4.4.4 Metadata search

In support of metadata search, the CCMS shall match queries against component metadata. For XML-based CCMSs, this matching may include metadata inside components.

Faceted search, often called faceted search navigation or faceted browsing, allows the user to search components through classification collections. Each collection represents a predefined facet on which the user may search.

When searching, the CCMS shall present the user with a list of facets. Each facet shall be presented with an indicator informing the user how many search hits belong to that particular facet with respect to the original search query. The user may select one or more facets, and the CCMS shall return results based on the intersection of the results from each facet.

12.4.4.5 Structured search

Structured search allows query targets to specific parts of the XML component structure using the XPath specification. Structured search is often combined with other forms of search to return highly specific results over large content sets. The CCMS shall provide a user interface to allow the submission of a structured-search query.

However, the CCMS should prevent the submission of malformed queries for malicious purposes.

EXAMPLE: A user searching wants to search only components that contain a source code example inside procedures:

//code[ancestor::task]

Or, a user wants to find each frequently asked question that also contains an image and an example:

//faq[image and example]

12.4.4.6 Saved searches

The CCMS should save searches so that the user may invoke the same query without having to enter it again manually.

12.4.5 Advanced search capabilities

The CCMS should support advanced search capabilities such as stemming, taxonomy support, and fuzzy matching.

12.4.5.1 Stemming

Stemming is the process of reducing search terms to their stem form. For instance, cats may be reduced to “cat”. An aggressive stemming system may reduce “argued, argues, and arguing” to “argu”.

12.4.5.2 Taxonomy support

Taxonomy support in search allows a taxonomic set of terms to influence how search is performed. The most common improvements to search are query expansion, search refinement, related search suggestions, and highlighting of related terms.

With query expansion, the CCMS shall analyze the search query and augment it based on the taxonomy to search on broader, narrower, or equivalent terms based on the situation.

EXAMPLE: Given the following taxonomy:

"Medical professionals"
"Nurse"
"Nurse practitioner"
"Doctor"
"Oncologist"
"Pediatrician"
"Cardiologist"
The user may search for “doctor” and the CCMS would expand the search query to also include “Oncologist”, “Pediatrician”, and “Cardiologist”.

With search refinement, the CCMS shall analyze the taxonomy and indicate to the user that there are narrower terms that produce a more refined search.

EXAMPLE: A search for “doctor” returns a large number of results, the CCMS suggests to the user to search on a narrower term such as “Oncologist”, “Pediatrician”, or “Cardiologist” from the taxonomy.

Similar to search refinement, the CCMS shall analyze the query and suggest other related searches based on the taxonomy.

12.4.5.3 Fuzzy matching

With fuzzy matching, the CCMS shall recognize terms that are not exact matches but similar enough to be considered relevant. Fuzzy matching is especially important when users spell terms incorrectly or use shortened versions of words. The degree of similarity of terms is often measured by the Levenshtein distance.

12.4.6 Versioning

The CCMS shall track version information for components in the CCMS. The CCMS shall maintain a version history for each component that tracks changes made to the content or metadata of that component.

The CCMS shall also maintain a local version number for components that may be used to return information about the committed change. The CCMS shall retain the following information for each committed version:

- User who committed the change to the component
- Time at which the change was committed

The CCMS shall:

- return the content of a component at a version in the component's history;
- return the content for a set of components at a specific time in version history;
- show the differences as insertions and deletions between the content of a component at two points in the component's version history;
- restore a component's content and metadata to a specific point in the version history.

12.4.7 Branch and merge

The CCMS shall allow the branching of a set of components. When branched, the CCMS shall create a clone of the original (sometimes referred to as its ancestor) components that shall exist separately from the original components. A clone reproduces the content, structure, and metadata of a component. Changes made to branched components do not affect ancestor components. The CCMS shall maintain relationships between the ancestor and original components so that an original component may be retrieved from a given ancestor component. The CCMS shall offer the same functions on branches that are available on the original content.

The CCMS shall allow a branch or a subset of a branch to be merged with its ancestor or a derivative of its ancestor. The CCMS shall also allow changes to be merged from an ancestor into a branch or a derivative of a branch. When merging, the changes in one component set shall be transferred to the other in both content and structure. Merging can result in conflicts that require manual intervention. The CCMS shall provide a user interface to resolve conflicts in merging.

Ideally, the CCMS shall provide a visual indication of conflicts/changes between files, both at the component and publication level and allow them to be accepted or rejected to complete the merge process.

The CCMS shall report on the last time a branch or subset of a branch was merged and what content was affected.

12.4.8 Release management

The CCMS shall allow concurrent working, future (or parallel), and past releases. Releases shall have the following associated metadata:

- the history of the given release, which shall contain a log of past or parallel releases which are ancestors of the current release;
- a name and target version number for the release.

A working release encompasses the unbranched content that is targeted for the next release of the publication. A working release may have a past release but it may not be a branch of a release. A past release is an archive of the publication and its components at the point in time when that publication was released.

As an archive, a past release is frozen, and as such the CCMS shall prevent changes to components in a past release. The CCMS shall instead allow a past release to be branched and modified and then merged or again released. A future or parallel release is a branch of a past or working release. The CCMS shall allow work on parallel releases concurrently with the working release or other releases.

12.5 Graphics and multimedia management

The organization shall specify how graphics and other media types shall be managed in the CCMS, in accordance with the capabilities of the CCMS. Graphics and other media types shall be maintained with the version control functions of the CCMS.

The CCMS shall store digital images, including graphics and other media types. Appropriate graphic formats that are widely used and are recognized industry standards, such as JPG, GIF, BMP, PNG, SVG, and others, should be supported.

The CCMS shall allow metadata to be applied to graphics and other media types to aid in search and retrieval. See Clause 12.1.3 for more details on metadata. The CCMS may use metadata embedded in digital assets to populate metadata fields.

After the text has been translated into target languages, the CCMS shall allow parallel, linked storage of the translated versions of the graphics files so that changes to the source language versions will initiate the re-translation of the target language version of the graphics.

The CCMS should store video and sound files in commonly used formats, such as WMV, SWF, MPEG, QTFF (.mov), WAV, MP3 and MP4.

Image formats with extractable text may be managed as part of translation packages provided by the CCMS.

EXAMPLE Text within SVG graphics files may be extracted and packaged for translation. SVG is an open standard created and developed by the World Wide Web Consortium.

12.6 Component Content Management System administration

CCMS administration includes the following activities:

- CCMS user administration
- Security and auditing
- Security provisions

12.6.1 Component Content Management System user administration

The CCMS shall allow for the creation, change, and deletion of a user and the users associated properties. A CCMS user is a person who participates in the creation, editing, and distribution of content.

The CCMS should:

- have a programming model and interface that is flexible enough that features can be turned on and off (or displayed and hidden) to provide a user experience tailored to the access level of the user;
- have a CCMS structure that is flexible enough to allow or deny access to components individually or by type;
- allow for the definition of access levels such that access to CCMS functions and components can be granted or denied to each access level;
- allow users to be assigned to or removed from access levels.

12.6.2 Security and auditing

Because some vulnerabilities allow unauthorized persons to run malicious code and thus take over the system or obtain access to private or proprietary information, the CCMS owner shall take precautions to protect the content and the users and to help prevent the system from becoming a host of malicious code or other attacks.

The CCMS shall provide levels of user authentication using methods such as password or biometric, and may use two or even more methods. If password or other information is stored for use in authentication, it shall be encrypted or cryptographically hashed or otherwise obfuscated.

The CCMS should log the dates and users involved in the following activities:

- User Login and Logout
- User Create, Change, and Delete events
- File Create, Change, and Delete events

Additionally, the CCMS should sort or filter on date, user, and event type and report the filtered and sorted results.

As with IT systems, CCMS security encompasses human access as well as machine-to-machine interfaces, including allowing remote control and monitoring. CCMs hardware and software are subject to security vulnerabilities. Additional security issues can be created through the way the live environment is configured.

12.6.3 Security provisions

The CCMS shall apply the following security provisions:

- provide a method to authorize users and securely manage associated sessions. If the system stores sensitive user information, such as passwords, they should be encrypted. Encryption should be used to secure transmitted user data;
- prevent users from elevating their privileges without re-authenticating;
- not transmit sensitive data in order to assist a user retrieving a lost or forgotten password or other personal information that may be used to compromise the security of the users account;
- provide encryption for sensitive data that may be exchanged between remote clients;
- provide sufficient and safe logging for unusual conditions, monitoring and alerting facilities to allow audit;
- enforce the roles, permissions and responsibilities of users as they apply to the CCMS;
- prevent third party applications, such as Web browsers and authoring tools used to interface with the CCMS, which may have their own security vulnerabilities, from granting access or privileges that may compromise the security of the CCMS.

NOTE: Controls as defined in ISO/IEC 27001:2013 should be used as guidance or normative requirements. The ISO/IEC 27000 series standards detail techniques for information security management.

Security designations or characteristics should be included when content is displayed or printed. The exact wording will vary in different organizations, and may have legal implications (which will vary by country). Typical security banners include the following:

- XYZ Corp. Confidential
- Internal Use Only
- Public Information

Pages without appropriate security designations may be implicitly public information (even though protected by copyright) or lacking in essential legal protections, depending on the legal jurisdictions from which they may be accessible. The security designation will not assure automated enforcement of the security designation. Declaration of security designation should not be considered sufficient to provide security control. CCMS design should include evaluation of passwords, encryption, and other techniques to provide additional security controls.

A qualified person associated with the CCMS owner should assess the adequacy of the security indicators and security protection for the system. Reviews should be at regularly scheduled intervals, as a result of a review-triggering event (e.g., system modification, change in ownership). A qualified person associated with the CCMS owner should assess the adequacy of the security indicators and security protection for the system. Reviews should be at regularly scheduled intervals, as a result of a review-triggering event (e.g., system modification, change in ownership).

12.7 Content authoring

CCMS can include authoring functionality as follows:

- General authoring
- Native authoring
- Authoring Integration
- Acquisition

12.7.1 General authoring

The CCMS shall provide a way to move content into the CCMS. Authoring is the process of creating new content. Acquisition is the process of moving existing content into the CCMS. CCMSs may provide native authoring functionality or integration with an external authoring interface. CCMSs may provide acquisition functionality.

12.7.2 Native authoring

If the CCMS provides an authoring interface, that interface shall:

- allow for the definition of content types;
- provide an interface for authoring components of the types that have been defined;
- guide or constrain the author, based on the definition of the information within and metadata around the content type being authored.

The CCMS shall describe the structure of the content type being authored and allow:

- creation of metadata elements to be applied to content types;
- addition of metadata elements individually to content types;
- display metadata elements during the authoring process;
- storage and retrieval of content objects between the authoring interface and the CCMS;
- deletion of components in the CCMS from the authoring interface;
- creation of a content hierarchy into which components can be added and support component-to-component links;
- creation of multi-paragraph text elements.

The CCMS should validate and enforce the structure to:

- allow the creation of independent metadata elements with controlled vocabularies that can be applied to one or more content types;
- apply metadata through drop down lists and other methods to consistently apply controlled vocabularies;
- support a variety of component organizational structures to be applied to components from the authoring interface;
- allow the creation of rich text elements that include block and inline semantic tagging (preferred) or format tagging (less preferred).

12.7.3 Authoring integration

Integration with an external authoring environment may be direct, where the CCMS projects its user interface into the external environment, or indirect, where the CCMS consumes the independently created output of an external environment. Indirect integration is considered acquisition as discussed in the following section. If a direct authoring integration CCMS is provided, the integration shall allow the author to:

- specify the target content type into which an externally authored component is to be stored;
- add additional metadata to the component created that is not supported by the external interface;

- guide or constrain the author based on the definition of the information within the content type being created;
- store, retrieve, and delete components between the external environment and the CCMS.

12.7.4 Acquisition

Acquisition is defined as adding existing content from an external source into the CCMS. The CCMS may include acquisition functionality. If acquisition is included, the CCMS shall:

- parse input into a single component, mapping the parts of the input to elements of the content type;
- log input parts that cannot be segmented or parsed and distinguish between mandatory and optional elements in the target content type;
- validate the components that are created;
- track the file name or query that constitutes the input content, the date and time of the acquisition event, and the identifiers of the components that were created and rejected during the event;
- fill additional metadata (not present in the input) with default values.

If acquisition is included, the CCMS should allow:

- input content to be targeted to content types in the CCMS (or put in the CCMS as non-categorized components);
- a single input to be segmented into multiple components;
- the specification, detection, and logging of useful exceptions during the acquisition event;
- monitoring, review, and configuration of acquisition events;
- an administrator to specify rules for how to fill additional metadata based on values that are present in the input.

12.8 Workflow

The CCMS workflow functionality includes workflow creation (or definition), workflow specification, and workflow reporting. A workflow is a named sequence of states that components of a particular kind are expected to go through from their original creation to their eventual retirement (by archiving or deletion)

12.8.1 Workflow functionality

The CCMS shall allow:

- creation of a single workflow;
- creation of a status (or state) metadata field with a controlled list of values that can be applied to each content object;
- administrators to apply the status field to content objects so that it is part of the metadata of each content object;
- authors to see and change the value of the status field in a native authoring CCMS if implemented or in a separate interface if native authoring is not supported;
- tracking the date and time for each status change;