

Content Management Using the Rational Unified Process

Part 1: Overview

Content management refers to a variety of tools and methods that are used together to collect, process, and deliver content of diverse types. The scope of content management is broad, and its challenges are many. This two-part article presents an overview of content management, particularly as it relates to delivering content on the Web, and examines why the Rational Unified Process (RUP) is the best framework for content management projects.

This Part 1 installment takes a high-level view, identifying the challenges of content management and the many activities it entails, and examining where those activities fit into the RUP. (Some previous familiarity and experience with the RUP is assumed.) Part 2 will be a roadmap explaining in further detail how to use the RUP to identify, plan, and manage the activities of a content management project.

Accompanying this article is a glossary of terms related to content management and a simplified case study that provides a context for examples (through an imaginary company named Acme). Sections of this article refer to corresponding case study sections where relevant.

Content Management Overview

Organizations increasingly turn to the Web as a means of distributing information, communicating with their customers, and seeking to differentiate themselves from their competition. However, simply uploading content to a Web site doesn't ensure that it will reach the proper constituencies or meet their information needs. The content must constantly be updated if it is to have ongoing value. It needs to be processed to ensure that it's ready for delivery to customers, and the delivery has to be designed such that users can locate what they need in a minimum number of steps. The content provided to each user should be matched to the needs of that user's business role — for example, technical documents to technical users.

The collection and processing of content intended for a Web site is a more complex and time-sensitive process than many legacy document management systems are designed to handle. A content management system may require integration with existing document management systems or may require its own separate workflow. There are many factors to consider.

So, while the *concept* of content management is not new, the *implementation* of a content management project is still a challenge, one for which there are more products than processes. Figure 1 shows an overview of content management from the process perspective.

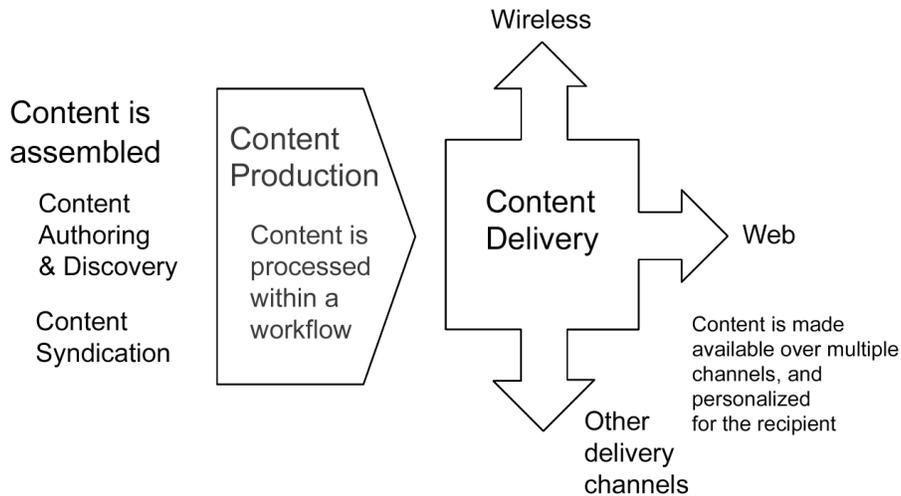


Figure 1: Content management process overview

Later in this article we'll look more closely at the content management aspects noted in Figure 1, but two key high-level points are worth mentioning here: the challenge presented by unstructured content, and the broad scope of a typical content management project.

The Challenge of Unstructured Content

The content management process starts with aggregating content from a variety of sources. Typically, some content exists in a **structured** form such as a database, but most of what needs to be delivered is **unstructured** content, such as articles, whitepapers, and marketing literature. This is especially true as a result of the Web's emergence as a delivery vehicle. (For more about the evolution of content management on the Web, see the section on that topic in the article "Content Categorization: An Orientation to Vignette and Content Management.")

Organizing and maintaining collections of unstructured content that continually change is one of the chief challenges of content management. A cursory examination of many content management products reveals that managing the workflow of activities related to unstructured content is an essential chore. The tasks within that workflow (some of which require human input) include:

- **Conditioning content** — This important early step involves preparing the content for entry into the content management system, including assigning keywords to facilitate categorization, searches, and personalization.

- **“Versioning” documents that have multiple contributors** — A document management issue related to unstructured content is that a content item may have multiple authors, or the editing process may require more than one person to be able to change a content item. So that additions and changes can be made without losing the work of any of the contributors, documents must be checked out and in. In addition, it’s desirable to have a simple procedure for rolling a content item back to an earlier version.
- **Integrating code and content** — Unstructured content items may be associated with particular templates for delivery (as may structured content; see Figure 2), or may need to contain or reference code (such as style sheets) that affects the delivery or behavior of the content.

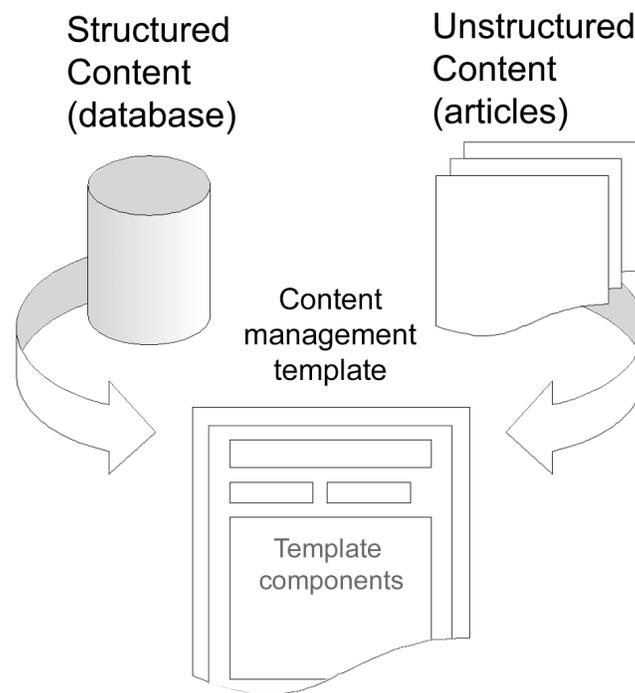


Figure 2: Templates for dynamic content delivery

The Scope of a Content Management Project

Content management, then, is much more than simply creating content and specifying that it’s to be published to the Web. The many activities a content management project entails include:

- assembling diverse types of content from multiple sources

- using **templates** to make the content available through a variety of channels (including wireless devices, e-mail, and fax as well as the Web), at the same time avoiding the need to create or manage redundant copies of the information in multiple formats
- providing content in multiple human languages
- using **workflow** (a predefined sequence of events) to administer the content throughout its lifecycle
- creating categories for cataloging the content
- tagging content appropriately so that users can easily find what they need, either by having it provided to them based on who they are or by having it recommended because of other content they've accessed
- identifying relationships and linking one piece of content to another
- automatically notifying users when content changes or when there are special business opportunities related to new or updated content
- integrating with other parts of an information system, such as knowledge management and document management

To summarize, content management is a broad undertaking that involves generating content of various types, delivering that content using applications that can meet (or even anticipate) the needs of various constituencies through a variety of delivery channels, and using workflow to administer the content.

The RUP and Content Management

Content management involves far more than buying or building a single application that manages content production workflow (which we'll call the **workflow application** in this article). A content management solution is made up of database, workflow, content delivery, and Web applications, developed as a group of related software projects. This makes it a perfect candidate for an established software development framework such as the RUP. Reasons for using the RUP as the framework for content management projects include the following:

- **Content management projects are software projects.** Content management projects have a lot in common with other software development projects, and therefore much of what they need to address falls within the RUP's existing disciplines. Using the RUP precludes the need to start from scratch. There's also a wealth of related guidelines within existing RUP roadmaps (such as the e-Business Solution roadmap).

- **Content management projects are complex.** A content management project typically involves a broad range of roles — business, technical, and creative — and it's important that the framework for such a project be able to encompass the diverse needs of those constituencies. The RUP helps strengthen communication among the various roles by identifying artifacts for them to share as they perform their activities. In addition to clarifying the vision and requirements of the project, these artifacts help identify risks that can then be addressed. The RUP is also well suited to handle the fact that parts of the content management solution (such as categorization and user interface) will most likely evolve over the life of the project, leading to ongoing iterations.
- **The RUP is a proven method with an installed base of over 100,000.** The need for methods that have been demonstrated to work continues to be rediscovered. Heroic effort does not scale, nor is it always successful. A configurable process that provides a common point of reference for the diverse roles and disciplines has been shown to offer the greatest success. The RUP has demonstrated success as such a process.

To better understand the activities of any complex process, it's useful to group the activities. In RUP terms, content management is a **discipline**, and groups of related activities within a discipline are called **workflow details**. Consistent with the process overview shown in [Figure 1](#), the workflow details of the content management discipline are:

- **Content development**, which addresses the standards, practices, and structure that govern the development of the content management solution, as well as the tools for dynamic content delivery (such as templates)
- **Content production**, which addresses the workflow of tasks that manage the content through its lifecycle
- **Content delivery**, which addresses getting personalized content to users over multiple channels

In the remainder of this article, we'll look more closely at what activities and concepts (and related artifacts) make up these workflow details and how they map to the RUP's phases.

The Content Management Discipline

This section elaborates on the function of each of the content management discipline's workflow details (content development, content production, and content delivery), providing general guidelines, defining related terms, and listing basic activities along with corresponding types of artifacts. It presents typical challenges that might crop up in each area, and refers to the Acme case study as an example; for an introduction to Acme and its content management-related problems, see the beginning of the case study.

Content Development Defined

Content development consists of establishing the framework of tools, languages, file formats, and guidelines that will be involved in creating or editing the content, as well as the definition of the content delivery means (such as templates). This workflow detail, which determines much of the architecture of the content management solution, can be broken down into the areas listed in Table 1.

	Activities and concepts	Artifacts produced
Content authoring	Writing text	Articles, whitepapers, marketing literature
	Content discovery	A list of possible content collections where existing content can be found
	Developing multimedia content	Sound files, multimedia files
	Developing content delivery templates	Code for delivery of dynamic content over the Web or other channels (such as wireless) Tested templates that reliably deliver dynamic content to all supported channels
	Addressing legal issues	Identified risks of publishing intellectual property to the Web Legal agreements stating site policies for use of content
	Internationalizing versions of the site for other languages and locales	Translated content and message catalogs
Defining development tools and guidelines	Selecting development platforms	Standards for site development, including supported code development platforms, databases, and markup languages

		An environment for testing and development using any supported platform
	Internationalizing development	Support for international character sets within the software requirements specification or supplementary specifications Multilingual versions of software development tools
	Selecting the development process	Adoption of the process and the tools for implementing it
Defining content structure	Defining an XML strategy	XML DTDs and related schemas
	Defining database requirements	A database schema and database application requirements
	Defining data conversion requirements	A data conversion strategy

Table 1: Content development workflow detail

The following subsections elaborate on content development, and examples of applying it to Acme’s scenario are provided in the corresponding section of the case study.

Content Authoring

In addition to writing text, content authoring activities include recording audio or generating other multimedia content types, creating templates for dynamic content delivery, and identifying or managing intellectual property issues such as copyrights or patents. In an international setting, translation of text to other languages falls in this area, as does the localization of data and time formats, sort order, and numeric representation.

Defining Development Tools and Guidelines

The area of development tools includes defining the environments within which various content types can be created, including markup languages, scripting or programming languages, and development platforms. It also addresses internationalization concerns, such as support for international character sets at the application, programming language, and database levels. Decisions regarding development tools are very significant to the application architecture.

Development guidelines describe the use of methods or modeling to define processes for development. This can include, of course, the use of the Unified Modeling Language (UML) and the RUP.

Defining Content Structure

The definition of data structures falls here, including XML document type definitions (DTDs) or XML standards, as well as database schemas and data conversions. The content delivery solution often depends on this architecture, making content structure design one of the highest priorities for content management.

Content Production Defined

Content production refers to the processing of the content in preparation for publishing it. Table 2 lists the areas covered by this workflow detail.

	Related activities and concepts	Artifacts produced
Using workflow to process content	Content routing	Analysis of how content is currently handled, including creation, delivery, and archiving Specification within the vision document of the content routing in the workflow application Rules for assigning content items to those responsible for working on them, as well as features within the workflow application to manually assign responsibility Appropriate features within the workflow application to meet

		<p>content routing requirements</p> <p>Use cases for all routing options</p>
	Task management	<p>A list of discrete tasks within the content production workflow, describing what they entail and what constitutes completion of each</p> <p>Use cases describing the various content management tasks</p>
	State management	<p>A list of states through which a content item might pass, criteria for the states, and tasks related to state changes</p>
	Role-based security planning	<p>A security plan listing the roles within the security structure and, for each role, the rights and constraints related to both content items and system administration</p>
Conditioning content	Categorizing content	<p>A categorization schema, with associated category keywords, and features within the workflow application to associate keywords with content items</p>
	Describing content	<p>Features within the workflow application to capture metadata for each content item, describing the subject, format, and other content conditioning information</p>
Content administration	System administration of the workflow	<p>System administration features for managing the creation and characteristics of users and groups, as well as the configuration of the workflow application and the rules governing it</p>
	Administration of content items	<p>Features within the workflow application to manage the state of content and provide for standard content management operations</p>

Aggregating content	Gathering content from various sources	<p>Descriptions within the software development plan of the content sources and the expected source format, transfer method, and delivery format for each</p> <p>Architectural plans to support communication among the various systems</p> <p>Consideration within the security plan for the receipt of content from outside locations</p>
	Content rationalization	<p>Strategies within the software development plan for collecting various types of content and presenting them as a cohesive whole</p> <p>Supporting templates for situations requiring content rationalization</p>
Syndicating content	Content conversion	<p>Template components for translating the data format, such as converting database query results to HTML</p>
	Subscription management	<p>A schedule for when updates are to be available, and considerations within the security plan for who can receive updates</p>
	Digital goods delivery	<p>Strategies within the software development plan for moving different types of content from a source to a destination</p>

Table 2: Content production workflow detail

The following subsections elaborate on content production, and examples are provided in the corresponding section of the case study.

Using Workflow to Process Content

The content production workflow focuses on routing content items to the people responsible for working on them. The routing should be rule-based (where the rules are based on use cases), with the option of manually assigning responsibility when required. The workflow is also concerned with providing the information and tools required to support each step of the content management project's lifecycle. It's broken into discrete tasks, which are then assigned using rules or administrative procedures.

It's advisable to implement role-based security measures to limit who has access to various content items and the changes they can make to those items. As in any security infrastructure, the use of groups facilitates granting and denying access to both the content and the administration of the workflow.

Conditioning Content

Conditioning content means adding information to or about a content item to facilitate searching and personalization (matching content to the needs of specific groups of users). It uses categories that are established for these purposes — specifically, using **category keywords** (terms used for matching search criteria about the content) and **metadata** (information that describes the content). Attaching metadata to content is also called **meta-tagging**, referring to the tags used to include information about a document in the head of an HTML document.

To describe the categories, there must be an accepted **taxonomy** (a division into ordered groups) and corresponding descriptive nomenclature. The taxonomy is used to create a category hierarchy, or tree, that can serve as a basis for a navigation scheme. It's not uncommon to have more than one way of describing the same thing, even within the same organization. Content associated with more than category should be able to be located by navigating from different directions.

Ideally, there's a means of rating the suitability or importance of content to its categories. An article describing how to install a network card, for example, might be associated with both installing hardware and installing drivers; a search from either direction would produce a hit. If the article discusses hardware configuration in depth and barely touches on drivers, a rating system could weight the categorization accordingly.

To enable content categorization to be automated, documents can carry metadata within them. Examples of document formats that support this are HTML and XML. In addition, a content management system can provide categorization for document types that don't have a way of carrying their own metadata.

Content Administration

Content administration encompasses the administration of both the workflow and the content items themselves. The needs associated with content administration include:

- submitting content items for publication
- locking content items to prevent access by more than one contributor at a time
- versioning content (assigning version numbers) as a method of tracking changes
- rejecting or approving content items for publication, and sending approved content to the content delivery system
- archiving content (maintaining copies of content items)
- deleting content (or marking it for expiration), or replacing it with content that's more current

One aspect of the content that's important to manage is its state. Examples of states include "incomplete," "requires additional editing," "ready for publication," and "requires approval of the Legal department." Once content has been processed by a task within the workflow, the state of that content might automatically change.

Aggregating Content

Organizations keep content in many places and in many forms, and often receive or subscribe to content from external sources as well (for example, from partner companies).

Aggregation is the assembly of content from a variety of sources, whether internal or external. Since it's unlikely that assembled content will all be in the same format, the following fall within aggregation:

- **Content rationalization**, which means collecting various types of content from multiple sources and presenting them as a cohesive whole (for example, bringing together a database of parts and prices and merging it on demand with Web page descriptions and pictures from another source, using a template)
- **Content conversion**, translating the format of the data (for example, converting information resulting from a database query to HTML for display in a browser)

Syndicating Content

Syndication means providing the most current content to browsers that have been configured to receive scheduled updates for the site, or providing content to other Web sites. It's increasingly common for companies to share information about their products and services with other companies for inclusion on their sites. In addition, client subscription technologies allow users to have their browsers automatically download the latest version of a page from a site.

Technologies that fall within syndication include:

- **Subscription management**, which involves the scheduling of updates and the security considerations of syndication (who can receive the content and how frequently updates should be performed)
- **Digital goods delivery**, which addresses the fact that not all content is simple HTML, and that the development plan needs to include multiple strategies for moving content from a source to a destination

Content Delivery Defined

Content delivery describes those parts of the content management solution that have the greatest public visibility: the Web site and other delivery channels. As shown in Table 3, it refers to the methods by which content is made available to users, including personalization (matching content to information that has been discovered or recorded about the user) as well as actual delivery. Since the performance of the site is an aspect of delivery that can be affected when content is provided dynamically, multiple-server solutions and caching are also included in the area of content delivery.

	Related activities and concepts	Artifacts produced
Personalizing content for users	Creating a personal experience for users	<p>Description within the vision document of personalization goals for the site</p> <p>Plans for gathering information about users explicitly or implicitly</p> <p>Defined roles for users, which play a part in determining the most appropriate content for delivery</p> <p>A categorization schema applied to the content items so that the proper content can be submitted to a particular role</p>
Delivering content over multiple channels	Making content available to various delivery media	<p>Description within the vision document of goals for multichannel delivery</p> <p>Templates that format the content for the medium over which the</p>

		user requests it
Performance and fault tolerance	Multiple-server solutions	<p>Description within the vision document of performance and fault tolerance goals</p> <p>A check of compatibility with platforms documented in the software requirements specification or supplementary specifications</p> <p>Methods for addressing personalization issues with multiple Web servers</p> <p>Methods for synchronizing dynamic content across multiple servers</p>
	Caching	<p>Description within the vision document of performance goals</p> <p>A caching strategy, including methods for identifying items to be cached to the content delivery server</p>

Table 3: Content delivery workflow detail

The content delivery requirements dictate application needs, which largely determine the standards established during content development. In fact, decisions made about the final delivery of the content drive most aspects of the content management project.

An elaboration of content delivery follows, and the corresponding section of the case study gives examples.

Personalizing Content for Users

Personalization means the creation of a personal experience for visitors to a site based on information gathered from them, whether by invitation and submission (explicit) or based on observation of their behavior (implicit). Again, there's a relationship between content delivery and other content management aspects, since the conditioning of the content (metadata and categorization) addressed within content production is based on the personalization design.

Delivering Content over Multiple Channels

A **channel** is a method of content delivery, the default method being the Web. Other methods might include e-mail, pager, cell phone, and personal digital assistant (PDA). Wireless Application Protocol (WAP) is an example of technology that's growing rapidly in some parts of the world. The increased availability of broadband access implies a growth in technologies, such as streaming media, that depend on high bandwidth for effective use.

Performance and Fault Tolerance

Given the complexity of content delivery applications, both performance and high availability are key concerns. **Fault tolerance** means the ability of a system to continue to operate despite the failure of some part of the system. The use of redundant servers provides fault tolerance, as well as performance gains when all servers are operating optimally. There are issues with server redundancy in content delivery systems that provide dynamic content, especially where personalization is concerned, and these issues need to be addressed in application and site design.

Another method of improving performance is **caching** — that is, keeping recently or frequently used pages in memory or on disk. The dynamic page generation commonly employed in content delivery increases overhead, and caching is a strategy for making the static parts of a page template readily available, reducing the time it takes to build and deliver a page.

Mapping Content Management to the RUP's Phases

In this section, we'll examine the workflow details of content management in the context of each of the four phases of the RUP (inception, elaboration, construction, and transition). The activities and concepts, and the resulting artifacts, listed above for each of the workflow details are mapped to the RUP's phases — a one-to-one mapping except for a few cases (marked in *italic*) where work on a particular artifact is split between two phases. As a result, the flow of a content management project emerges.

Where applicable, evaluation criteria from the RUP are presented here, but these don't include all the criteria for assessing the phases of a project using the RUP. The emphasis is on those criteria that are most applicable to a content management project.

Inception Phase

The goal of the inception phase is to achieve concurrence on the lifecycle objectives for the project. Table 4 shows which aspects of a content management project come into play during this phase. Primarily, it's important at this time to define and record the vision of the project in the vision document. For content management, the vision typically focuses on content delivery, with some interest in content production driven by those responsible for the content production workflow. Further additions to the vision document are made during the elaboration phase.

Activities and concepts	Artifacts produced
Content discovery	A list of possible content collections where existing content can be found
Addressing legal issues	Identified risks of publishing intellectual property to the Web
Selecting development platforms	Standards for site development, including supported code development platforms, databases, and markup languages (<i>exploration</i>)
Creating a personal experience for users	<p>Description within the vision document of personalization goals for the site</p> <p>Plans for gathering information about users explicitly or implicitly</p> <p>Defined roles for users, which play a part in determining the most appropriate content for delivery</p>
Content routing	<p>Analysis of how content is currently handled, including creation, delivery, and archiving</p> <p>Specification within the vision document of the content routing in the workflow application</p>
Task management	<p>A list of discrete tasks within the proposed content production workflow, describing what they entail and what constitutes completion of each</p> <p>Use cases describing the various content management tasks (<i>start</i>)</p>

State management	A list of states through which a content item might pass, criteria for those states, and tasks related to state changes
Making content available to various delivery media	Description within the vision document of goals for multichannel delivery

Table 4: Lifecycle objective milestone

Related evaluation criteria as applied to a content management project are:

- **Concurrence on scope definition** — Content development involves establishing the project’s software scope and boundary conditions, including an operational vision and acceptance criteria: what’s intended to be in the product and what’s not. In addition, concurrence on the scope of content delivery is critical (since content delivery has the highest visibility within content management projects), as is agreement on what it will take to accomplish the desired results.
- **Requirements understanding** — Content delivery and content production will depend on defining the critical use cases of those systems. The primary scenarios of operation will determine the major design trade-offs to be made. Exhibiting (and maybe demonstrating) at least one candidate architecture against some of the primary scenarios can facilitate architecture decisions. In addition, investigating the needs of the content production system may include preparing the supporting environment for the project.

Elaboration Phase

The goal of the elaboration phase is to baseline the architecture of the system, providing a stable basis for the design and implementation efforts during the construction phase (see Table 5). It should be demonstrated that the baseline architecture will support the requirements of the system at a reasonable cost and in a reasonable amount of time.

Activities and concepts	Artifacts produced
Selecting development platforms	Standards for site development, including supported code development platforms, databases and markup languages (<i>selection</i>) An environment for testing and development using any supported platform
Selecting the development process	Adoption of the process and the tools for

	implementing it
Multiple-server solutions	<p>Description within the vision document of performance and fault tolerance goals</p> <p>A check of compatibility with platforms documented in the software requirements specification or supplementary specifications</p>
Caching	Description within the vision document of performance goals
Internationalizing development	<p>Support for international character sets within the software requirements specification or supplementary specifications</p> <p>Multilingual versions of software development tools</p>
Creating a personal experience for users	A categorization schema applied to the content items so that the proper content can be submitted to a particular role
Gathering content from various sources	<p>Descriptions within the software development plan of the content sources and the expected source format, transfer method, and delivery format for each</p> <p>Architectural plans to support communication among the various systems</p> <p>Consideration within the security plan for the receipt of content from outside locations</p>
Categorizing content	A categorization schema, with associated category keywords, and features within the workflow application to associate keywords with content items
Content rationalization	Strategies within the software development plan for collecting various types of content and presenting them as a cohesive whole
Subscription management	A schedule for when updates are to be available, and considerations within the security plan for who can receive updates

Defining an XML strategy	XML DTDs and related schemas
Defining database requirements	A database schema and database application requirements
Defining data conversion requirements	A data conversion strategy
Content routing	Rules for assigning content items to those responsible for working on them, as well as features within the workflow application to manually assign responsibility Use cases for all routing options
Task management	Use cases describing the various content management tasks (<i>complete</i>)
Role-based security planning	A security plan listing the roles within the security structure and, for each role, the rights and constraints related to both content items and system administration
Digital goods delivery	Strategies within the software development plan for moving different types of content from a source to a destination

Table 5: Lifecycle architecture milestone

Critical during the elaboration phase is the need to set up the supporting environment for the project, including creating a development case and guidelines as well as setting up tools. The initial setup of a workflow application, if one has been purchased or is being given a trial, occurs during this phase. To allow prototyping of templates, at least one development Web server needs to be put in place. If the automation of the content delivery system is incomplete, content that has passed through the test workflow can be manually copied to the server for testing.

The criteria worth noting for this phase are:

- **Stability of the architecture** — This is evaluated through one or more architectural prototypes (for example, templates for dynamic content delivery).
- **Addressing the major risks** — The architecture evolves out of a consideration of the most significant requirements (those that have a big impact on the architecture of the system) and an assessment of risk. Because of its complexity, content delivery will present some of the greatest risks.

Construction Phase

The focus of the construction phase is to clarify the remaining requirements and complete the development of the system based on the baseline architecture (see Table 6).

Activities and concepts	Artifacts produced
Writing text	Articles, whitepapers, marketing literature
Developing multimedia content	Sound files, multimedia files
Developing content delivery templates	Code for delivery of dynamic content over the Web or other channels (such as wireless)
Content routing	Appropriate features within the workflow application to meet content routing requirements
System administration of the workflow	System administration features for managing the creation and characteristics of users and groups, as well as the configuration of the workflow application and the rules governing it
Administration of content items	Features within the workflow application to manage the state of content and provide for standard content management operations
Describing content	Features within the workflow application to capture metadata for each content item, describing the subject, format, and other content conditioning information
Content rationalization	Supporting templates for situations requiring content rationalization
Content conversion	Template components for translating the data format, such as converting database query results to HTML
Multiple-server solutions	<p>Methods for addressing personalization issues with multiple Web servers</p> <p>Methods for synchronizing dynamic content across multiple servers</p>
Making content available to	Templates that format the content for the medium

various delivery media	over which the user requests it
Internationalizing versions of the site for other languages and locales	Translated content and message catalogs

Table 6: Initial operational capability milestone

To accelerate the development activities that take place during the construction phase, it's desirable to achieve some degree of parallelism in the work of different development teams. The proper use of content management templates, and in particular the reuse of components within templates, can help toward this goal.

The primary criterion for this phase is whether the product release is stable and mature enough for deployment. The flow of content from content production to content delivery needs to be stable by the end of this phase. The content delivery application must be able to assemble content dynamically and deliver it to the channels that pass it on to users. Before this project can be handed off to the transition team in the next phase, the progress of the organizational transformation must be far enough along that the content production workflow has become part of the routine of the workers involved.

Transition Phase

The focus of the transition phase is to ensure that the site is available for end users (see Table 7). This phase can span several iterations, and it includes testing the site and making minor adjustments based on user feedback. At this point in the project lifecycle, user feedback should focus mainly on fine-tuning (for example, on the effectiveness of personalization and ease of use). All the major structural issues should have been worked out much earlier in the lifecycle.

Activities and concepts	Artifacts produced
Developing content delivery templates	Tested templates that reliably deliver dynamic content to all supported channels
Addressing legal issues	Legal agreements stating site policies for use of content
Caching	A caching strategy, including methods for identifying items to be cached to the content delivery server

Table 7: Product release milestone

Transition phase tasks will be discussed in greater detail in Part 2. For now, note that when the RUP is applied to content management, the transition phase tasks are essentially the same as they would be for any other RUP-managed software project. The primary evaluation criterion for this phase is whether the users are satisfied.

Summary

Content management covers a lot of territory, from defining the need to provide some type of content, to creating and delivering the content (including creating, configuring, and administering the delivery mechanism and translating the content for different delivery channels). This article has presented the three workflow details of content management — content development, content production, and content delivery — along with related definitions, objectives, and concerns, and examples in the form of a case study.

We've also looked at some reasons why the best framework for content management projects is the Rational Unified Process — a proven method and multidisciplinary approach to identifying and assigning activities and responsibilities as a project is iteratively delivered — and at how content management activities and concepts map to the RUP's phases.

Part 2 will examine using the RUP for content management in more detail, providing additional guidance for requirements definition, identifying tasks and roles, and introducing specific content management artifacts for the RUP. Where appropriate, it will present more detailed descriptions of activities within the case study as examples.

Related Resources

- “Content Categorization: An Orientation to Vignette and Content Management” by Damien Kennedy (Rational Developer Network article). In addition to discussing the evolution of content management on the Web, this article describes a hierarchy of categories for classifying the many aspects of content management.
- [W3C's HTML Web site](#). Here you'll find information on the history of HTML and the Web, as well as links to other sites containing information on the evolution of the Web.
- *The Rational Unified Process: An Introduction, Second Edition* by Philippe Kruchten (Addison-Wesley, 2000).

Copyright 2001 Rational Software Corporation. All rights reserved.