



Electronic Content Management 101:

What is an electronic content management system?

How do I implement it in a regulated environment?

How can I support my users after implementation?

How can a Business Analyst help me with all this?

What is content management?

- From Wikipedia, the free encyclopedia:
“Content management, or CM, is a set of processes and technologies that support the evolutionary life cycle of digital information. This digital information is often referred to as content or, to be precise, digital content. Digital content may take the form of text, such as documents, multimedia files, such as audio or video files, or any other file type which follows a content lifecycle which requires management.”

What this really means...

- Managing content can no longer be thought of managing large volumes of paper documents
- Content management systems:
 - Contain content files that are of different formats and which at some point can become related to one another
 - Make use of taxonomies to define content

The reality is that...

- As part of the normal course of business, the typical life science organization creates, reviews, updates, and publishes a tremendous amount of content
- All this content is generated for a reason - whether its for external review (i.e., submittals to the FDA), or internal use (i.e. SOPs)
- Regardless of its ultimate destination and purpose - this content somehow needs to be properly maintained and controlled

The challenges this creates...

- In growing life science organizations, certain business processes tend to evolve when the end product is the traditional paper document
- Such business process can lead to:
 - Lack of standardization (i.e., authors making custom format decisions vs. focusing on core information)
 - Different authors re-creating the same or very similar information in different documents
 - Copying and pasting the same information into different documents - which now has to be stored and maintained in multiple places

What needs to be done ...

- Develop a content strategy where all involved departments understand the ultimate goal, and work in solidarity with each other in order to achieve this goal
- Understand the current processes being used to create content
- Determine what content needs to be created - and who creates, reviews, approves, and ultimately uses it
- Determine how this content can be broken out into discrete, reusable elements, or “chunks” that can be authored once and reused many times in different document “assemblies”
- Create standard templates for these content chunks
- Developing authoring standards so that the focus for authors is on creating content they are responsible for (vs. format)
- Define the future business process based on these new rules

The benefits this provides ...

- Better defined authoring roles:
 - Definition of the overall assembly structure
 - Definition of who is responsible for authoring individual content chunks
 - Definition of templates for each chunk
- Facilitates a consistent content creation process - authors can focus on content and not on spending time with developing (inconsistent) formats
- Lends itself to reuse and ultimately the consistent, predictable production of an end (document) product

How this can be achieved...

- By using an Electronic Content Management System, which can:
 - Encourage the up front development and enforcement of consistent document authoring rules
 - Facilitate single sourcing & reuse
 - Maintain document relationships
 - Implement granular levels of security
 - Versioning, searching, workflow, etc

The payoff for users

- Easier to locate documents
- Documents can be accessed globally
- Users access documents based on role and document status
- Previous versions can be accessed
- Electronic inbox and notification
- Document audit trails

Questions for the business

- What percentage of time are authors spending on tasks relating to presenting content vs. simply entering content?
- What percentage of content is similar? Can common "chunks" of content be re-used?
- What percentage of content has to be re-entered, and subsequently re-reviewed, re-approved, and re-published?
- Are there ever instances where content that is updated in one document requires updates to other documents that contain very similar or associated information?
- Is there ever a need to publish content into multiple output formats or media and to multiple languages - all from the same source file?

What is XML?

- XML stands for Extensible Markup Language
- It is an open standard technology recommended by the World Wide Web Consortium (W3C)
- It is used to describe the structure of content, and not its format
- It's not dependent on any one platform or technology solution
- It facilitates robust communication between different business applications
- It enables the delivery of the same source file to multiple outputs

Benefits of XML to the business

- An organization's ability to reuse content can have substantial business benefits.
- When using XML, authors no longer have to worry about document formatting - as they can focus their efforts solely on entering and referencing content.
- Since XML separates the content from the format, you can reference the same re-usable content in multiple documents without actually updating the core content of either the referenced or the referencing document.
- Therefore, if referenced content needs to change, an author can make the update in one place. This will in turn enable the change to automatically proliferate to wherever its referenced.

Benefits of XML to the business (cont.)

- Limiting review and approval to only the changes made ultimately produces more consistent content, saves time in the authoring cycle, and helps reduce the risk of errors.
- Business users can realize the benefits of increased productivity, as they will be able to more effectively retrieve and reuse content that can ultimately be reused for multiple outputs.
- A well defined content structure allows business applications to automatically exchange information. This enables the automation of many processes that are currently being done manually by business users. It also greatly improves the ability to exchange content between different information systems.
- Therefore, XML offers the potential to make the business process wrapped around managing regulated content shorter, easier, and more reliable.

Benefits of XML to the FDA

- The FDA is faced with mounting pressure to: 1) review and approve an increasing amount of drug submission information 2) ensure that drugs which are either unsafe or ineffective are kept off of the market, and 3) ensure that consistent and accurate product information is available to both physicians and end consumers
- Given the numerous benefits of XML, in recent years, the FDA has started to rely on XML-based standards - both in terms of submission structure (i.e. eCTD) as well as content (i.e. SPL) to facilitate the management of regulated content
- XML can: 1) allow product information to flow from the manufacturer - to the FDA - and ultimately to the end consumer in an efficient and reliable manner, and 2) ensure that this information is consistent and accurate
- By using XML-based standards, the FDA can reduce the exchange of information down to what's changed. This will help to speed the approval of drugs and streamline updates to the National Library of Medicine.



Some terminology...

For those of you who are new to this...

Repository

- A repository contains:
 - The content files
 - Content file templates
 - The information about the content (metadata)
 - Content object hierarchy
 - Business rules (i.e.. Lifecycle information)
 - Workflow templates
 - Definition of users and groups

Cabinets and Folders

- A Cabinet:
 - The “topmost” folder in the hierarchy
 - Contains folders and/or content files
- A Folder:
 - Contains other folders and/or content files
 - Can be linked to more than one cabinet
 - A content file can be linked to more than one folder

Versioning

- Checkout:
 - Content files can be “checked out” by users in order to “lock” the file for editing
 - File can be updated only by the user who has it checked out
- Checkin:
 - Stores updates to the content file made by the user to the repository
 - Removes the “lock” from the file

Version Tree

- Users can create multiple versions of the same document through checkout and checkin
- A version tree consists of all the file's versions (stored as separate objects)
- The content management system assigns both implicit (i.e. version number) and explicit version labels to each version in the version tree

Lifecycle

- The “status” of the content file
- For example:
 - Draft (authoring stage)
 - In review (review process)
 - Reviewed (review process complete)
 - For eSignature (signature process)
 - Approved (signature process complete)
 - Effective (visible to user community)

Searching, Viewing, & Access

- Content files can be searched by both metadata as well as the contents of the file itself
- The content management system will provide the user interface that facilitates searching
- User can then view the metadata and or content of located document depending on their permissions (access level)

Renditions & Annotations

- Renditions are:
 - Associated with native content files
 - Typically XML or PDF documents
 - Differ from the original only in format
 - PDFs can contain watermarks and/or overlays
- Annotations are:
 - “Sticky-notes” that reviewers can attach to a PDF rendition
 - Are typically stored separately from the rendition (when using web pages)

Virtual Documents

- Powerful feature which allows individual content object “chunks” to be related together into one “container” object
- Versioning and security can be applied to both the individual object chunks as well as the virtual document parent object
- Individual nodes within the virtual document structure can be “fixed” bound to a specific version, or bound to the most current version of an object “chunk”

Workflow

- A workflow is the representation of a business process, whereby the tasks associated with document(s) follows a specific, pre-defined route (from user to user)
- Serial workflow: each task can be processed by one user at a time, and the user must complete the task before it progresses to the next user
- Parallel workflow: multiple tasks within the workflow can be processed at the same time



Implementing your system...

Hughes IT Consulting, LLC

First, define your SDLC

- An SDLC (System Development Life cycle) should always be followed when implementing systems that are subject to cGxP regulations
- Following the principles and practices outlined in the ISPE's (International Society for Pharmaceutical Engineering) GAMP (Good Automated Manufacturing Practice) guide is probably the best place to start
- What this really comes down to is using good quality practices to implement software that meets the needs of your business

5 reasons software projects fail

- Lack of sponsorship
- Lack of up front definition
- Lack of implementation methodology
- Lack of quality focus
- Lack of success criteria

The SDLC – making it work

6 basic steps:

1. Planning
 2. Requirements
 3. Design
 4. Development
 5. Testing
 6. Operation and maintenance
- Key: It's all about making sure you follow a good process!

Step1: Planning

- What business goals are we trying to meet?
- What does the business consist of today?
- How can we improve our existing processes?
- What processes should we automate?
- Perform feasibility study for automation
- Define and document the draft “to-be” process
- Develop your implementation plan
- Get project sponsorship!

The Implementation Plan

- Purpose of the project (i.e. why are we doing this?)
- Project goals (i.e. what do we expect to achieve?)
- Project scope (i.e. how will this be accomplished?)
- Stakeholders (i.e. who will benefit?)
- Feasibility study for automation
- Assumptions
- Constraints
- Risks
- Critical success factors (i.e. how can success be measured?)

Step 2: Requirements

- User requirements – what the business user wants the system to accomplish, (given the current possibilities of information technology, their business goals, budget, etc)
- Functional specification – a definition of how an information system will meet the user requirements (i.e., design objectives of what the system will do)

Why document user requirements?

- User requirements help bridge the gap between the business and IT
- What managers want is not necessarily what business users need
- What COTS (“Commercial Off the Shelf”) vendors sell is not necessarily what business users need
- Knowing what your users want gives your business a competitive advantage during vendor selection
- Requirements define what should be tested and verified before the system is in production
- Either invest time now defining user needs, or later on struggle with a product that doesn’t fit their needs
- The cost of knowing what your users want up front is far less than learning later on!

Why document a functional specification?

- It verifies that the system will have all of the required functions necessary to meet the user requirements (it should be traceable to the user requirements)
- It should be written in a way that both the user and the vendor understand it
- Why use a vendor's software if we find that it cannot meet all of the user requirements?
- Its better to make sure that we have chosen the right software up front before purchasing and implementing it into production!

Example: User vs. Functional

- User: “When choosing to view a document, the system must display a PDF rendition of the document to the user.”
- Functional #1: “When a non-PDF document is created and saved in the repository, the system will queue it for automatic rendering.”
- Functional #2: “The rendering server will periodically poll the queue for documents that are to be rendered.”
- Functional #3: “Once the native document has been rendered, the PDF rendition will be saved to the repository and associated with the native document.”

Step 3: Design Specification

- Will consist of 3 main components:
 - Required hardware (Hardware Design Specification)
 - Required software (Software Design Specification)
 - System architecture – how the hardware and software will work together

Hardware requirements

- Main infrastructure specifications should be defined in a Hardware Design Specification
- System storage specifications (i.e. how much content will system need to hold?)
- Required peripheral devices (i.e. any required controlled printers, etc)
- Operating Environment (i.e. where the equipment will reside)
- Backup and recovery solutions
- Hardware acceptance (IQ) verifies installation of hardware components

Software requirements

- The Software Design Specification must define how the software will meet the functional specification
- System software components (i.e. what database?, web server?, mail server? ... etc)
- Products - list specific products or modules that will be implemented
- Version – list specific versions of vendor's software
- Platform – what base OS is required to run software

System Architecture

- Define system inputs
 - Content created by the user/system owner
 - External system interfaces (list all inputs from external systems)
- Define what is happening to the content once it's been created
 - What is being done with the content (i.e. automatic rendering, watermarking, etc)
 - Workflow (i.e. electronic signatures, etc)
- Define system outputs
 - What will the end user ultimately see?
 - Reporting tools for the business owner

Why do I care about design?

- It serves as a blueprint as to what the vendor(s) will need to configure during the Development Phase
- It gives you a precise definition as to what this system really consists of
- It is essential for change management
- It helps assure that all of the requirements are being addressed

Step 4: Development

- If customizations are necessary, be sure that coding standards have been identified up front before development begins
- Ensure that code reviews are conducted to enforce adherence to these standards, and make use of code review checklists
- Ensure that custom software modules are properly identified in the source code
- Ensure that custom software modules are traceable to specific functional requirements

Step 5: Testing

- Testing should minimally be done at 3 levels:
 - Unit testing - to be done by application developer
 - Functional (OQ) testing - the system was built and configured according to the functional specification
 - User (PQ) testing - verification that system meets all of the user requirements.
 - PQ testing is an effort to prove that the system will fit into the newly defined business process. Idea is to simulate (as much as feasible) how the business will actually operate with the new system before accepting the software for actual production use.

Why so much testing?

- If changes need to be made, then they can be done up front before putting the system into production.
- Depending on the test results, adjustments could potentially roll all the way back up to the business process, user requirements, functional specification, and design (yet another reason to define all of this!)
- Its less costly and less painful to get it right up front before the system is used in production!
- Key Point: If you are in the business of making widgets, then you better make sure you can produce a real widget with your new process and new system before committing to the system for production use!

Risks of not following the process

- Commercial off-the-shelf software (COTS) is typically thought of as a way to save time and money on expensive custom programming
- But, if not properly managed, it can have potentially disastrous results
- What may look nice during a demo may crash in production (i.e. functional issues) or cause the users to struggle with the system (i.e. usability issues) or has capabilities that don't fit with your business (i.e. process issues)

OK, its live now...

- What about support?

Define a best practice for support

- Like your implementation plan, your support plan should be identified up front during the planning phase of your project
- Different models/guidelines exist (i.e. GAMP guidelines for keeping the system in a validated state, etc)
- The important point is to identify a good process, document it, make sure everyone understands it, and be disciplined about consistently sticking with it

Need to assess current state

- If your organization already has a standard for IT governance in place, then you'll need to understand that and determine how the support for your application will “bolt on” to that standard
- If one does not exist, then determine what is currently in place, what you can inherit, and what you'll need to implement that's new

How do we get started...

- If, after doing your current state assessment, you've determined that certain aspects are missing that is beyond a project level - you'll need to determine if any departmental initiatives are in place to fill the gap
- Next, determine what project specific support processes need to be developed in order to "bolt onto" the existing processes that are already in place
- Use the elements of your support model as a guideline for what you need to develop
- Make sure that all key stakeholders (i.e., Help Desk, Infrastructure, etc) participate in developing the plan and ultimately agree to what the process will be

Who can help me with all this?

- As you can see, implementing a content management system is a BIG undertaking
- An experienced Business Analyst can be a key asset to ensuring successful delivery and facilitating ongoing support

What is a Business Analyst?

- According to the International Institute of Business Analysis (IIBA):
 - “A business analyst works as a liaison among stakeholders in order to elicit, analyze, communicate and validate requirements for changes to business processes, policies and information systems. The business analyst understands business problems and opportunities in the context of the requirements and recommends solutions that enable the organization to achieve its goals.”

What a “BA” often does ...

- A Business Analyst wears many hats
- The BA is a key focal point who participates in everything from requirements gathering, analysis, design, installations and system testing to working with project managers, users, IT infrastructure, software vendors, and application developers
- The Business Analyst helps analyze and define business needs
- Serves as a liaison between your business and information technology solutions/vendors
- Determines what solutions can or cannot do to help run your business
- Can facilitate the implementation of software solutions as well as the definition of support agreements with vendors
- Must be an advocate of *your* business, and not for any particular software vendor.

Questions?

- Feel free to contact me at:

hughesitconsulting@gmail.com