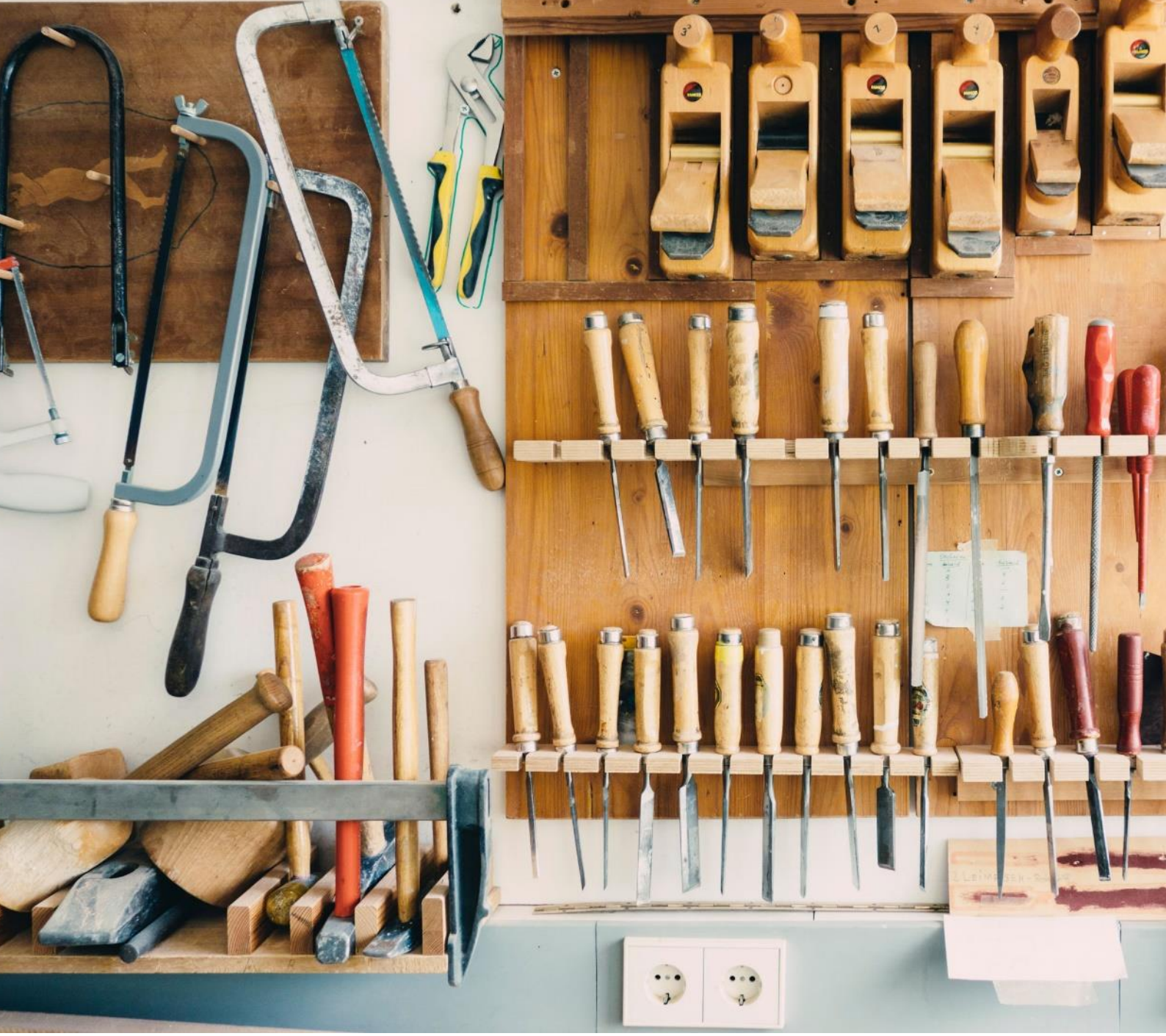
Routing Digital Content Throughout The Enterprise

Envision, plan and deploy a well-connected, manageable and sustainable content infrastructure

Provided by Picturepark Content Systems
Authored by David Diamond

Table of Contents



Overview

Vendor-based software discussions typically pull conversations in directions that put vendor products at the center, leaving readers wondering how to adapt the information to their own real-world circumstances.

This document aims to provide a discussion of enterprise content creation, management, routing and measurement that comes from no specific perspective, and promotes no specific technologies, disciplines or industries.

While a number of the individual concepts and recommendations herein can be handled by "out of the box" software provided by Picturepark and other software makers, your first take-away from this document should be:

No out-of-the-box software solution will enable you to do with your enterprise content exactly what you need to do today and what you expect to be able to do tomorrow.

For this reason, it is important to not shop for a ready made content management "solution" as much as you shop for a *means* to a solution for *your* organization's content management needs. Part of this will be software and part of it will be people you trust to perform the research required and build and maintain systems and policies that make sense, and are sustainable.

Software is not a solution; software is a tool. What you do with your software tools can lead to a solution or a mess. The difference between the two outcomes is most often dependent on an organization's willingness to take the time and effort to discover and understand its own needs, and to be able to translate those needs into reasonable expectations of the digital, financial and human resources available.

More directly, *is what you need to do reasonable, given reality?*

Don't expect this to be a quick process. There are many moving parts in enterprise content management. No one person knows them all, and no one software platform can manage them all. And while you might consider yourself a content management novice now, don't be surprised to find that the expertise you acquire during the process of making this happen for your organization turns you into an expert.

Users and Flow

Before taking on the details of your content management practices or content system design, it helps to think in the abstract about two key considerations:

- Who or what will be creating, editing and consuming your content?
- What rules and regulations will govern the creation, editing and consumption of your content?

Accurate and complete answers to these two questions will enable you to determine what you need from your content management initiative.

People and Machines

Though human users might be the most obvious creators, editors and consumers of content, these roles are increasingly shared with machines. (When we speak of machines in this document, we refer to software and hardware systems, collectively.)

It's a good idea to think of both types of "user" when planning or expanding a content system, even if you can't currently imagine a use case for one type or the other.

Human Users

Factors to consider for human users include:

- Experience
- Motivation
- Requirements
- Location
- Device

These factors enable you to create user personas that will help you design your content systems. A user persona is a brief profile that describes these factors for a typical type of user.

An example persona follows the details sections that come next.

Experience

If all your users will be experienced pros who are familiar with your content, policies and business practices, you can make assumptions about knowledge when choosing and configuring your content systems.

For example, help documentation you author for experienced users might provide guidance on how to use a given interface widget, without explaining the "why" of using that gadget. This is the sort of documentation one would expect with a medical device designed for use only by trained physicians. An electronic thermometer might require some explanation of features, but there is no need to explain to doctors why or when to take a patient's temperature.

Experienced users will appreciate brevity and directness in the information they consume.

Novice users, on the other hand, need more guidance. If your content systems will be used by interns, "newbie" employees, partners, customers or other external users, your audience will include less experienced users. You must consider this in your system selection and configuration, and in the documentation you provide.

In virtually all cases, human users require some level of training. Expect startup training and ongoing training to keep brains fresh and increase system and policy knowledge. But never lose sight of the differences in experience levels when designing training programs.

Motivation

Motivation is an often overlooked consideration, but it is an important one.

Which best describes your user base?

- Users are eager to access and use content
- Users are required to access and use content

Users eager to use your content systems will be far more forgiving of complexity than those who do so because they have no other choice. If your policies and procedures are so cumbersome that few people understand them, you'll find that less eager users will complain more, use the system less, and provide no meaningful input to improve things.

On the other hand, if you provide a system that features human touch points, policies and procedures that are enjoyable and reasonable, you might convert some of those "required" users into eager advocates.

Requirements

"Find content" is an expected requirement for anyone who will use your content systems, but this is not a granular enough description to assist you in system design.

A salesperson, for example, might need to "find approved content for use in presentations." This description tells you two important things about this requirement:

- The content must be approved
- The content must be suitable for use in presentations

In many cases, a given user group will have several different requirements for the system. For example, managers might need to access content statistics to create usage reports, but they will also need to access images or other content they can use in those reports.

Location

Though physical location is less a consideration than it once was, there are still things to consider about where your users will be located.

First off, if you will have users outside your corporate networks (which is likely), you will have to consider access and security options. In some cases, users will have to connect via virtual private networks (VPNs) or other mechanisms your IT team requires for network access from external locations. While these technologies can certainly improve security, they can also increase headaches for users.

User experience problems can be exacerbated when some of your content systems are within your corporate network and others are in the cloud, unaffected by the security software your IT team uses. It might make total sense to an IT manager when it is and is not necessary to first connect to a VPN before using a given software system; but to the average user, the difference is simply a bookmarked URL that sometimes works and sometimes does not.

Before getting too far with your system design, speak to someone in IT who can provide answers to the following:

- What is the process for connecting to the corporate network? If the process is too cumbersome, users will use the system less and complain more.

- Will all components of your content systems work properly when users connect via the mandated security technology? In some cases, network differences or permissions can result in problems.
- Are any of the content system components you are considering, or data you intend to store therein, in violation of any security protocols, mandates or data protection regulations?

Another location consideration is performance. Users who are farther from your core systems, or those who are connected from locations where fast access is not available, might experience performance problems that make some parts of the system unusable.

Device

Finally, consider the types of devices users will want to use with the system. Desktop computers are great for certain purposes, but mobile devices are better for other purposes and use cases.

When considering devices, think not just in terms of which devices users *prefer* to use, but think in terms of use cases. For example, if your business involves people contributing content they create at remote locations, such as photos of real estate or sporting events, do not assume desktop computers will be the device of choice for these remote users.

The significance of thinking in terms of use case is that you will find that some components of your system will require mobile-friendly access, while others will not. For example, those editing video or making large system modifications might tell you that mobile interfaces make them less productive.

Finally, keep in mind that gone are the days of supporting specific operating systems, at the expense of others. Your users will connect using Windows, Mac OS, Android, iOS, Chrome OS, Linux and any number of other variants. It is a good idea to support them all.

Fortunately, this is much easier these days than it has been in the past because most software interfaces are browser based. If a content system component you are considering requires special software be installed on users' computers, consider carefully what this means for your users and those who will support those users.

Persona example

Using the factors presented above as a guideline, we can draft the following persona.

Scott is a marketing professional in his 20s. His experience with institutional policy and his knowledge of technology are limited. He is capable of learning, but he would rather not be required to learn too much about content systems because he doesn't see this as his primary job responsibility.

Scott uses his phone to find and share content across social media. He neither contributes nor edits content. He does not have the authority to approve content.

With this persona defined, you can speak simply of "Scott" when referring to any user who fits this profile.

The value of “Scott” is that you know a few things from reading his persona profile:

- Don’t expect him to understand complicated configurations or policies
- He will not have interest in mastering the system
- He requires reliable mobile access to find and share content
- All content Scott sees should be approved for use by another

Repeat the exercise for other types of users you expect to support. Make your personas only as detailed as is required to illustrate the differences between user types. Depending on your circumstances, personal attributes, such as race or political affiliation, might be irrelevant.

When configuring or expanding your system, now or in the future, you will be able to connect your intentions and plans to user personas. If you find yourself planning to do or add something that does not seem to benefit any persona, ask yourself whether the addition is really necessary.

Software development often uses personas to guide development efforts. When building an enterprise content system, you are developing software, so the same rules and best practices apply to you.

Machine Users

Non-human content creators, editors and consumers are already more common than you might think. The roles machines play in content is not usually identical to those held by human users, but machines’ roles are equally important.

Examples include:

- Creation (translations, memes, banner ads, observation and reporting)
- Validation (spell check, structure verification, completeness)
- Enhancement (auto-tagging, image modifications and restoration, relating, summarizing)
- Consumption (sharing, embedding)
- Control (trigger, respond, combine, forward, notify)

Perhaps the world’s largest machine “user” of content is the Google search engine. Google reads content, evaluates that content, indexes the content, and provides links to the content via search operations (sometimes to other non-human users).

When you pull up weather reports on your phone, you see another example of automated content creation, editing, consumption and publishing.

Obviously, your phone is not performing weather analysis. And you can rest assured that there is no human manually taking weather observations, adding sunshine and rain graphics, and then sharing them to some location where your phone finds them. It’s all system-to-system communication.

Stock quotes come to you in the same way, as do the flight delays and “time to leave” notifications that Google Assistant provides.

It might seem futuristic to think in terms of machine users of content, but this is nothing new. Printers have been machine *consumers* of content for decades. Scanners and digital cameras have been machine *creators* of content. Spelling and grammar checkers have been machine *editors* of content.

Even the notion of machine-based content routing is not new. Email servers and network switches have been routing content for a long time. Save, copy and share operations also route content, albeit to locations chosen by users.

What is relatively new is the concept of machines working with content in accordance with rules and other parameters that you or other systems define—automated, without human interaction.

If this concept scares you, pick up your phone and check the weather. You will see that the world does not come to an end just because a bunch of systems are connected to one another to help you decide whether you will need a jacket tomorrow.

User Groups and Roles

Virtually all enterprise software systems are designed to “think” in terms of groups or roles, rather than individual users.

Groups and roles greatly simplifies system administration and security, because global access changes can be quickly managed, without having to update the accounts of what could be hundreds of thousands of individual users. More specifically, groups and roles can be aligned with policy definitions, which helps ensure the system is configured as it has been designed on paper.

Groups and roles are as applicable to machine users as they are human users, though you will likely create different groups or roles for each type of user.

Groups combine users based on some attribute of who those users are, such as employees in Marketing or Sales, or partners, customers, etc. *Roles* combine users based on what users need to do in the system, such as upload content, edit content or manage the system itself.

In the roles paradigm, a given user is typically added to more than one role, collectively defining all rights and duties that user has within the system, regardless of the corporate department to which that user belongs. For example, someone who contributes content to the system might also need to find and share content.

Consider the following example roles:

- **Content Contribution**
- **Content Browsing**
- **Content Sharing**
- Content Editing
- Content Management
- Content Archiving

By adding a user to the three boldfaced roles, she can perform the functions she needs, without being able to edit, manage or archive content. Note that these roles have no bearing on a user's department or title, or even whether that user is an employee of the company. For this reason, a single role structure can be used to manage employees, freelancers, agencies or anyone else.

In some cases, you might find that your groups or roles align with your personas, which is an added benefit of having personas. Referring back to the "Scott" persona, any user who fits that persona would be added only to the Content Browsing and Content Sharing roles.

What is important to consider when looking at content system components is how each works with groups or roles. Though each component might come from a different software maker, the permissions paradigm they use must be compatible, or else system configuration and maintenance will be much more complicated, if not impossible.

For example, if one component of your entire system uses only user groups that must be mapped to the groups in a global user directory, such as LDAP or Active Directory, you will be unable to adopt a roles-based approach to your greater user management. (Your global user directory is more likely organized by department than content considerations.) This is another reason to befriend someone from your IT team who can guide you during content system evaluations.

Policy-based Content Flowpaths

After thinking about users, roles and groups, think in terms of the path content travels throughout your organization:

- From where or whom does your content come?
- Where does your content need to go for processing, editing, approvals and storage?
- How will your content be used, by whom and from where?

You will likely need to create multiple content flowpaths because situations and needs will differ for different types of content. The granularity with which you create these flowpaths will depend on how they will be used, and by whom. The good news is that you do not need to start with too much detail.

Start by defining a few high-level content types, such as:

- Financial documents
- Product/service content
- Policy documents

The value of starting with broad-strokes categories like these is that they apply in virtually all cases. For example, the above categories are as applicable to a local doctor's office as they are to Apple and Siemens. In time, you can further refine your core categories, if needed.

Apply the questions above to the lifecycle of your financial documents:

From where or whom do your financial documents come?

Your financial documents might be produced in house, or they might come from an external accounting or other provider. The source of these documents should be considered because that source will need access to your system in order to submit the documents.

Where do your financial documents need to go for processing, editing and storage?

Once submitted, what happens to the documents? Perhaps they are sent for review by managers or your CFO, after which they are sent to your CEO for final approval. Maybe national or institutional regulations mandate that these documents be stored on dedicated storage devices not accessible to most users.

How will your financial documents be used?

What happens to those documents once they are approved? Perhaps they are shared across your organization, sent to media outlets and investors, regulatory bodies and partners, and maybe even published on your website, or shared via social media. Once the financial cycle has ended, the documents might be archived to a special location where they are no longer editable.

In answering these questions, you create a checklist of sorts that you will use for further system, policy and procedure design:

- What types of business systems will you need to manage the flowpath?
- Which users or user groups are required, optional or should be excluded?
- What network or security concerns have you identified?
- What approval processes are required?
- What regulatory adherences and reporting are required?
- What storage requirements are unique to this content type?

Finally, and perhaps most important, is whether you can identify any "holes" in the flowpath.

For example, if policy mandates that your CFO must approve a document before your CEO sees it, what happens if your CFO is unavailable?

If policy mandates that a given financial filing be submitted to a regulatory body within 10 days of the close of a quarter, what mechanisms do you have in place to ensure compliance with this policy?

What happens if one or more of these processes fail?

What happens if the system goes offline or is compromised?

Try to interview those at your organization who have institutional knowledge about the content type and the requirements of the process. These folks can help you account for things you might not think to consider, and they can help you define what should happen when things go wrong.

Then do the same exercise for your other content types.

Where you find yourself thinking “it depends on the type of content,” you need to get more granular about your flowpath definitions. In some cases, an entirely new flowpath will not be required, if you can cleanly isolate differences into branches that are easily defined and configure into your software systems.

For example, you might find that the flowpath for your regulatory filings differs from that of your quarterly investor briefings only in in that the investor briefings are not distributed to the public.

The goal is that you document from where content originates, what happens to it, and what course of action must be taken if something goes wrong.

Content Creation and Acquisition

At the start of each content flowpath will be a content source. Your initial flowpath sketches can identify nonspecific sources, like “internal staff” or “agency,” but you will eventually want to get more exact with regard to actual content sources, where possible.

Defining content sources with more granularity enables you to:

- Budget for actual purchase and configuration costs
- Understand and plan for technological limitations and incompatibilities
- See where “off the shelf” software will work, and where custom development will be required

A selection of common content source types follows. You might have additional sources.

Applications

The most common sources of content at many organizations are desktop applications, such as Photoshop, PowerPoint, Autocad and other applications that are installed on users’ computers. In these cases, users create content and save those creations on their computers, or on remote file servers or file sharing services.

In other cases, cloud-based applications, such as Salesforce, Eloqua or Marketo, can also be used to create content, though these applications are more geared toward working with content created elsewhere. One popular example of cloud-based tools with which content is created is the Google App suite.

The “onramp” from these applications to your larger enterprise content system will be via integrations or preferred (or mandated) storage locations.

Integrations between these tools and a component of your content system enable application users to directly access shared content, or store newly created content in the content system.

Integrations can be convenient, but they can also cause trouble. As applications are updated, there is a chance an integration will cease to function properly. The vendors of host applications do not always announce changes reliably, so the makers of integrations can be caught off guard. This, in turn, can leave users wondering why what worked yesterday no longer works.

When a specific storage location is used as the entry point to the greater system, users must be trained to know where that location is, when to use it, and whether they are required to do anything special with the content they submit, such as adding metadata values or naming the files in a certain way. If users do not know the rules, they can end up contributing incomplete or unsuitable works to the content system.

As a rule, integrations are preferable when there is a value to storing or accessing shared content before the user’s document is complete. For example, an InDesign page layout designer benefits from having direct access to shared content for placement into layouts.

User Curation

When you bookmark a website or a YouTube video, you might not think of this as content creation. But if your content system permits you to add metadata to that link to define it and give it context within the scope of your interests, you are creating content.

Some content systems enable users to paste links that are then used as the basis for new content records. In some cases, previews of the content might be provided.

For a given research project, you might have teams that scour the web for links to content that will be useful to your design, marketing or sales teams. Collectively, though you've stored only links, those links become valuable contributions to your greater content system. As with content you create in house, these links might be shared, commented upon, and referenced in other materials.

Just as music fans have grown accustomed to streaming content that they do not possess, it will become increasingly common for organizations to assemble collections of content published elsewhere, yet provides unique value to internal teams, once enhanced with relevant metadata.

Automated Content Creation

As mentioned, automated content creation is on the rise. In most cases, this is not about machines feeling inspired to create illustrations or articles on their own; instead, it is about machines using data to make decisions about pieces of existing content that can be combined and rendered or presented, usually based on a formatting template.

Companies with large product catalogs are familiar with automated creation of product listings and advertising elements. When a new product is released, existing templates can be used, which saves considerable time. In some cases, the new content is created and made available to those who need it, without any human intervention.

If there is no direct API integration between your content system and the application that renders automated content, your best option might be to have your content system periodically check a dedicated folder or server location to pick up new content that is saved there.

Stock Houses (Licensing)

Content licensed from external sources must be handled carefully. In most cases, the licenses under which you acquire content are limited. For example, they might permit use in one region, but not others; or they might permit single use, or online use only.

If the content is used outside the bounds of the license, it can be a costly mistake. The goal is to make clear to all who will have access to the licensed content in what context it may be used, for how long, and which attributions are required.

The best way to make this clear is via metadata directives, and the best time to add those directives is when the content is added to your content system. The goal thereafter is to make sure that no matter where throughout your content system the content travels, those directives remain clear and can be machine read, if needed.

Other considerations unique to licensed content are the license extensions or renewals for content that is not licensed in perpetuity. Most professional content systems offer some means for enabling you to add reminders or checks for expirations; but it is worth inquiring about how that works, if you plan to license content—especially if those licenses will differ from one another.

Freelancers and Agencies

Content that comes from freelancers might also involve license restrictions that should be noted and honored. But more specific to freelancers is whether or not you offer them access to your system.

In some cases, freelancers post the content they create in their own locations and send links to those locations to those who should access the content. But if your freelancers will contribute their works directly into your content system, you will need to make sure they have access and that the content they contribute properly identifies the source and any usage restrictions that might be in place.

Systems that provide “named” accounts, which means each user has her own account, typically show the name of the user who contributed the content to the system. But if your system uses no named or individual user accounts, you will need to make sure there is some mechanism—most likely via the use of metadata—for the contributor to identify himself. Otherwise, you can end up with a lot of new content for which there is no record of ownership or origin.

Upstream Business Systems

An additional source of content might be upstream business systems, such as product information managers (PIM), master data management systems (MDM) or some other system into which users input data intended for use across the organization.

These systems might be connected to your content system via direct API integrations, but data can be also transferred by scripts or other tools that copy data between the systems. In these situations, the source system exports data to a file that the script or tool then transfers to the content system, where it is ingested.

In many cases, these data transfers happen nightly. If there is a need for data to be transferred instantly or depending on complex rules, a more sophisticated solution is required. Though the export/import method can certainly be easy to configure, you might find that it lacks options to account for errors, such as when one system is offline or data is not entirely processed or complete.

Depending on the data and how often it is accessed, it is possible that days, weeks or months of erroneous data could be transferred before anyone notices the problem. As bad a problem as “bad data” is on its own, the problem worsens when other “downstream” systems rely on that data.

For example, say that product data flows from your PIM system into your content system, from where it is incorporated via templates into ads that are made available to media outlets. Let us further say that a product manager erroneously lowers a price from 99.99 to 9.99.

A properly designed integration could flag pricing changes that exceed a given percentage, ensuring the data does not get published before an authorized individual can verify the change is correct. In a simple data transfer, the error would likely remain unnoticed until customers started calling en mass to purchase the great deal.

The export/import option is most common when a business system lacks an application programming interface (API), making direct system connections difficult. This is typical where older systems are used, but when data transfer requirements between two system are very simple, organizations can choose to take the easier route, rather than enduring the cost and complexity of building and maintaining a custom integration.

Content Management and Collaboration

Once content is in the system, it must be managed to ensure it can be accessed by those who need it, and to ensure it cannot be accessed by users and systems that should not see it.

Among the most common aspects of content management are:

- Controlling access
- Storage and archiving
- Collaborative communication
- Adding real-world metadata
- Validating and improving automated metadata

- Creating semantic links

Controlling access

The most common types of access are:

- See the content (browse, find and preview)
- Edit the content (make changes or add new versions)
- Access the content (share, download, print)
- Manage the content (move it or delete it, change access permissions)

Each content system differs in how it offers controls over content. Some systems enable you to define unique access to each piece of content, while others require that you assign controls to containers in which content is stored, or to permissions templates to which the content is assigned.

There are advantages to both methods.

When you can adjust access permissions for a given piece of content directly, this is very easy to understand—you give access to those users or machines you want to have access. If someone says she needs access to a given piece of content, you can simply connect to the system and add that user.

The drawback to this method of permissions management is sustainability: If you have 100 pieces of content and 10 users, this is relatively easy to manage. But if you have a million pieces of content and 100,000 users, this method of control would be unmanageable.

Instead, enterprise content systems tend to rely on associating user (or machine) groups or roles to containers or templates that can be applied en mass. This approach applies a layer of abstraction that makes management easier: Instead of potentially millions and millions of permissions matrix possibilities, you map a given set of templates to a given set of user roles.

In order for this control method to work, you not only need a system that is based on user groups or roles, as previously discussed; but you need to define policy for each content type that defines who should be able to do what with that content, and when.

In turn, you configure containers or templates to reflect that policy. When a given template is assigned to a given piece of content, the content inherits the permissions defined by the template. If the template is updated, those changes are reflected for all content associated with the template. A million pieces of content can be updated with a single change, which is the primary value of this approach. Another advantage is that you can more easily imagine the access assigned to a given piece of content just by knowing to which template is has been assigned.

In most content systems, permissions are applied either *additively* or *exclusively*. Some systems offer both approaches.

In an additive system, a given user's access is defined by the combination of templates and rules applied to a given piece of content. For example, being part of the Marketing group might not grant a given user access to financial documents; but if that same user was also part of the Senior Managers group, and that group does have access to financial documents, then that marketing user would have access too.

In an exclusive system, typically only one template or container is active at a time on any given piece of content. The permissions defined by that template are in effect for associated content, no matter how many different roles a user plays in the organization.

As a rule, a system that is roles based should also feature an additive permissions model to provide the greatest flexibility.

Some additive systems also enable you to create templates or containers that impose exclusive permissions, when applied. So, for example, if there were a NO PUBLIC ACCESS template configured to behave exclusively, it would override any public access granted by other templates applied to the content.

Keep Access Simple and Policy Driven

As a rule, it is a good idea to provide only the minimum level of access that any given role requires do what it needs with the associated content. For example, if your external partners will not be contributing content to the system, do not grant them that permission.

You might think there is no harm in permitting trusted partners to add content if they want, but the greater issue is one of policy adherence: If your policy that defines the External Partners role does not include the option to contribute content, and you enable this permission on that role, you are defining roles that are outside the scope of policy. Later, if there is an access problem in the system, it can be much more difficult find and correct if system managers can't rely on current configuration to accurately representing written policy.

Exclusive templates can be important time savers when configuring permissions, and they can also serve as safety nets to ensure content is not available when it should not be. Good examples are press releases or new product information that is under embargo, or content for which licenses have been lost or that should be archived. Without the option of an exclusive template, a content manager would have to manually review the entirety of templates associated with the content and remove those that granted access. This is a lot of extra work, and it can result in unintended consequences.

Worth noting is that when content is shared for, say, social media access or editing by an external user, a "tokenized" permission is added. This permission override is what makes the content accessible to others who are not working from within the content system. In the case of a request to edit, the permission might be temporary; in the case of a share to social media, the tokenized permission might be left in place indefinitely.

Storage and archiving

An important content management concern is storage space: where, what type and how much. In most cases, these choices will be up to you. In some cases, content systems mandate where and how they store content. More modern systems will not impose such limitations.

If your entire content system will reside in the cloud, alternative storage options might be limited, and your concern about those options might be limited at first too. But it is still important to know how your service provider is accounting for storage, and disaster avoidance and recovery.

You might decide to have multiple types of storage. For example, faster storage for current and popular content, and slower, more affordable storage for older or less frequently accessed content. In addition, you might have offline (and offsite) storage options that are used for archived content that should no longer be readily available, or just to ensure you have a more secure backup of your entire system.

Offsite backups provide organizations with a means to safeguard their content collections from disasters, such as fire, earthquakes, floods or cybercrimes that might destroy online data.

Data Protection Regulations

Data protection regulations might require you make clear to users the locations in which their data is processed and stored. In some cases, not all countries, regions or jurisdictions will be legal options. (See GDPR.)

Though there is understandably extra cost involved in maintaining an always-current offsite backup, the value of doing so will depend on what a "disaster" would mean for your organization. If, for example, the complete destruction of your content system would mean the end of your organization, or it would result in costly litigation with customers, you might find the cost of offsite backups to be worthwhile.

Other considerations are *recovery time objective* (RTO) and *recovery point objective* (RPO) policies. RTO defines the acceptable amount of time that data or systems can be remain offline and inaccessible after a failure. RPO defines the acceptable amount of time (or traffic) that can exist between data backups.

These policies can vary dramatically, depending on the organization. For example, a company like Amazon cannot remain offline for long without suffering millions in lost revenue. And imagine the outcry if the world's previous day of Facebook activity was lost forever, even if the system itself went offline for only a minute or so.

Though it is easy to say that systems and data cannot be inaccessible for more than a minute, and that you require that no data be lost in the event of a system or network failure, this requirement is both unreasonable and unaffordable.

When a system has gone offline because of an external network glitch, it might be back up again within minutes, with little or no effort on the part of your IT teams. But if the failure is the result of your hardware (or personnel), recovery will not be so easy.

Well trained IT teams are ready to replace hard drives and other hardware components, but even the fastest swap will require time. In some cases, a system can take longer to reboot up to operational status than it took to swap the faulty component. And there is the testing required to ensure everything is functioning properly. And all of this, of course, assumes your team was able to identify the problem quickly.

Cloud service providers sometimes offer different RTO/RPO options to clients; but rest assured, "fast" will not be "cheap." Worse, the warranties offered by hosting providers might not adequately cover your losses. So, even if you can calculate what downtime costs you per minute, you are not likely to recover those losses via service warranties.

"Cold storage" options are popular for organizations that have large content archives. This type of storage can cost a fraction of traditional storage. A number of cloud storage providers

offer cold storage as a service. If your chosen content system does not directly support the cold storage provider you prefer, see if access can be developed via the content system's API. There are many cloud backup services available that should meet your requirements, some of which you might have already entered into agreements with for other purposes.

Depending on the capabilities of your system, content might be moved between storage types automatically. So, when something is accessed once, it might be copied to faster storage where it remains cached for a given period of time. Or, when a user marks a piece of content as being "archived," the system might move it to archive storage.

Similar in concept to using multiple storage types is using a content distribution network (CDN). These networks cache recently used files so that they are faster to access for future users. Increasing their value, CDNs can distribute content to locations all over the world, so that Internet latency is reduced, and system downtimes or peak loads can be compensated.

In virtually all cases, the use of a CDN is not apparent to users and managers of a content system. In most cases, the use of one is merely a factor of subscribing to the CDN of your choice and enabling support for it, either from within your content system or at the CDN control panel.

Managing Remote Content

Even if your entire content system is used for nothing more than management of remote links, some storage will be required. Thumbnails, cached previews and metadata all require storage space. Keep this in mind when creating storage policy and making storage decisions.

What is important to keep in mind about CDN use is that it might affect the usage statistics managed by your content system. For example, the first time a person accesses a piece of content, the content is copied to the CDN, from where it is accessed thereafter for a period of time. But it is possible that the actual number of accesses via the CDN will not be available in your content system. If accurate statistics are a requirement, make sure you discuss this with the makers of your content system to see what options are available to you. As for all storage and processing of data, make sure that your chosen CDN complies with your regulatory requirements.

If your content collection is large, or will become so over time, make sure your system supports multiple types of storage, and the movement of content between storage locations. Just this one aspect of content management alone can improve performance for users and greatly reduce operational costs.

Collaborative communication

If your content production or management should include collaborative interaction between employees and, optionally, freelancers or agencies, you'll need some means for facilitating that communication.

There are a few times when collaboration can be most useful:

- Content planning
- Content production and revisions

Content planning

At some organizations, decisions about what content should be created or otherwise acquired come from team discussions. There might be multiple such teams in an organization. One might include folks from Marketing, while another includes people from the documentation department.

In most cases, these discussions will take place outside the content system, often in more specialized creative collaboration or project management tools, some of which might be provided by the authoring or creation tool directly. But the content system can also be useful to content teams while they are making their decisions:

- Permit teams to easily determine what content has already been produced
- Enable teams to accurately determine which existing content is popular and appreciated
- Provide teams with a placeholder for directives for content production or procurement

Those first two points might seem obvious: *Enable people to search and see statistics*. What is important there are the words *easily* and *accurately*. Will content teams know what to search for in order to determine whether a piece of content has already been produced or procured, or is in the planning stage with another content team? Do statistics show numbers in the context of trend, time and other factors? Or do they just show total numbers of downloads since the content was added to the system?

In this capacity, the content system serves as a research tool to planning teams. Speak to those teams and ask them how they now come to decisions with regard to what is needed. Then, make sure the content system can deliver on as many of those needs as possible.

The placeholder concept is valuable because an empty content record is the start of the proposed content's lifecycle. In some cases, an empty record could define a wish list content item; in other cases, a content placeholder can be created for content that is expected, such as quarterly filings and annual reports for the next decade.

A placeholder not only enables others to see that a piece of content has already been planned for, but it enables planners and designers to provide development directives and schedules. This, in turn, enables content creators to prioritize and plan more in detail in other production or collaboration systems. For example, if a content development directive comes in, and the placeholder record shows that the content must be created in a week or else it will not provide any value, development teams can use this information in their planning.

Additional value would come from the content system having connections to other planning systems, such as product releases or marketing campaigns, so that content teams could see what was coming.

Some content systems have built in communications tools. In some cases, they might be as basic as email notifications between users; in other cases, complete workflow engines guide users through planning and production.

Though the idea of a workflow engine can seem attractive, the value of such a tool depends on one very important factor: How well can you “templatize” your content planning? In other words, if you do not have a consistent method through which content is planned, it will not be easy to translate that into an automated workflow.

This concern carries through to the production of content.

Content production and revisions

Extending many aspects of content planning is content production.

As mentioned, placeholder records can become production directives that provide content and scheduling guidance. But they can also serve as locations for content-specific collaborative discussions.

Worth noting is that many creative tools have built in discussion or annotation features. These features might be valuable at times since they enable content creators and reviewers to discuss, decide and make changes while the production of the content is ongoing. Once resolved, many of these discussions need not to be considered again.

Examples of comments that would be added during such creative reviews include:

- “Move this logo further from the photo.”
- “Remember to save this with lossless compression.”

Examples of the types of comments you’d want to keep independant from the creation process include:

- “We will need this content approved for use by the end of the month.”
- “Contact the Japanese distributor for guidance about an alternate hero image for the Japanese version. The chosen image could be considered culturally insensitive for that market.”

Comments like these explain directives and strategic decisions that affect the creation and use of content. These are the kinds of comments that will be valuable for managers to see later. But artists benefit too, because directives and change requests are always found in a single location.

After content has been available for a while and someone sees that an update is required, that person will know where to go to start the process, even if the first step is merely to ask other managers if they agree that an update is required.

Contrast this to a production cycle in which email is used to communicate, and people are required to know where to find review files, to whom comments should be sent, and where all this information is archived.

When using a single system as a production directive and commenting hub, you are able to have a master record that is easily accessible to everyone who should see it for each piece of content you produce or otherwise acquire.

- Why was a different graphic used on the Japanese signage?
- Why did we get a rush charge to have these graphics produced during that time frame?

Questions like these will be answered by looking at the discussion archive. Being able to see what has been suggested and why it was or was not accepted as an idea can save teams from wasting time doing and trying things over and over.

Adding real-world metadata

Those familiar with digital asset management understand the value of accurate and complete metadata profiles. In short, if content does not include quality metadata, that content is as good as lost.

When you type a search term in a content system, you are asking the system to match that term to available metadata profiles. In fact, the actual content associated with the metadata profiles is rarely, if ever, considered in search operations. When content is of a type that contains text, such as documents, it can be helpful to consider that text in search operations;

but in most cases, metadata matches are given priority because metadata can lead to more reasonable and expected search results.

There are a few reasons why metadata profiles are more valuable than content when it comes to search:

- Visual content, such as images or video, might contain no words
- Textual content can be too abstract or varied to provide meaningful search results
- Metadata profiles can be designed to improve search, or extend data found in the content

That last point is key. When designing metadata profiles, you can account for the ways in which you expect users to find content. By creating metadata fields and values for those purposes, you can provide textual content that is weighted more heavily in search results.

Consider the content you are now reading. If you already knew of this document and you were trying to find it in a content system, you might use words from the title, such as “routing content enterprise,” or something similar. But if you were looking to discover new content similar to this document, and you had no idea that this document existed, there is no guarantee that you would think to search for these terms.

Without knowing the title of a given document, you would likely search for terms you expected to be within the content. The problem with the terms *routing*, *content* and *enterprise* is that they are found within many documents. Worse, the documents that contain the terms might have little or nothing to do with routing content through the enterprise.

This is why metadata profiles are so important. They enable you to define the gist of a given piece of content, using terms users are likely to use.

For similar reasons, metadata is important to search results from third party search engines, such as Google. Though Google is primarily a content search engine, it does consider metadata that is associated with content in certain ways. This can be expected to become even more valuable over time because Google (the company) is a proponent of *linked data*, sometimes called *semantic metadata*.

Linked data can be thought of as connections to external information that help define a given object. For example, if Google were to see that a content item called, “The Smartest Little Phone” was linked to the topic “children's books,” Google could surmise that the content was not about compact mobile devices.

Metadata can also provide historical information about content. Though this could, of course, include actual historic information about photos of ruins, and such, it also refers to the history of the content itself: When was it created? When was it last updated? When is it due for another update?

Historic metadata can also refer to licensing or the planning or production discussions mentioned previously.

The current status of content is also represented by metadata. Simple statuses, such as “In Draft” or “Approved for Use” can be used to let users know what they can and cannot use. But status values can also apply to considerations like popularity, regional interest and more.

Most commonly, metadata is used to describe content itself. An image contains a “flower” or it was taken during “daylight” or in the middle of “summer.”

Users typically think first of descriptive metadata when performing searches. But when faced with too many results for a given search operation, they start to think of options for refining those results. This is why the value of a complete and accurate metadata profile for each piece of content is so important.

Even better if the content system supports flexible metadata profiles. Picturepark content systems, for example, support a technology called *Adaptive Metadata*. The value of this is that each piece of content in a system can potentially have a totally unique metadata profile. While in practice, this is not usually the case; it is common for a Picturepark system to support many different profiles, depending on the types of content in use in that system.

For example, content related to an event might call for metadata values that are unique to that type of content:

- Event name
- Event date
- Event location
- Sponsoring department

It would make no sense to include metadata fields for these purposes on, say, your financial or product documents. Yet, in systems that do not support flexible metadata profiles, this is what you would have to do if you wanted to include this information on your headshots.

Adaptive Metadata also permits metadata profiles to change over time. This is valuable, for example, when using the technology to provide extra fields for production or review. When the fields are needed, they can be added by any permitted user. When they are no longer needed, they can be removed by any permitted user.

In making metadata profiles so flexible, users find Picturepark systems less complicated because they see on each piece of content only those metadata values that make sense.

Validating and improving automated metadata

Automated metadata tagging is becoming more common in content management. The premise of such a technology is alluring: Send an image or video to a service and have it send back a selection of tags that describe the content.

This can save countless hours of human tagging when used with content and use cases that are suitable for automated tagging. When automation is not suitable, what is returned from the tagging service can be humorous at best, or misleading at worst.

Three important things to consider before adding automated tagging to your content system are:

1. Have you tested the auto-tagging using your actual content?
2. Do you have an approval workflow in place to verify automatically added tags?
3. Have you legally reviewed the terms of such service auto-tagging provider?

Only by sending your own content to the tagging service can you properly value the results you will get back. Among the things to look out for are the tags themselves — are they accurate? In addition, do they provide the level of detail that your users will need?

With generic or stock content, it might be enough to say that an image contains a “man” and a “tractor,” perhaps throwing in the tag “green” to describe the tractor. But if your business is building green tractors, “green tractor” won’t likely be descriptive enough to suit the needs of your users.

For example, terms that describe a tractor’s purpose, engine horsepower or lift capacity would be common search criteria, but no automated service is likely to be able to provide these values.

You could decide to add these tags manually, but then you have to factor in the value of the core automated service. Did it save much manual effort? Did it introduce errors that had to be corrected by a human?

Approval workflows are common during content production, but they are rarely used for metadata editing. While everyone has an opinion about the way an image looks, people usually assume someone else will be able to tag it adequately.

But to ensure metadata accuracy across your content system, you must have some means for verifying (and potentially overriding) any automated metadata service, such as auto-tagging.

Ideally, any erroneous tags users remove, along with new ones they add, will be fed back to the auto-tagging service to help it “learn.” This not only enables the service to become more valuable for use with your specific content, it can prevent the same wrong tags from being assigned in the future, which will save your users time.

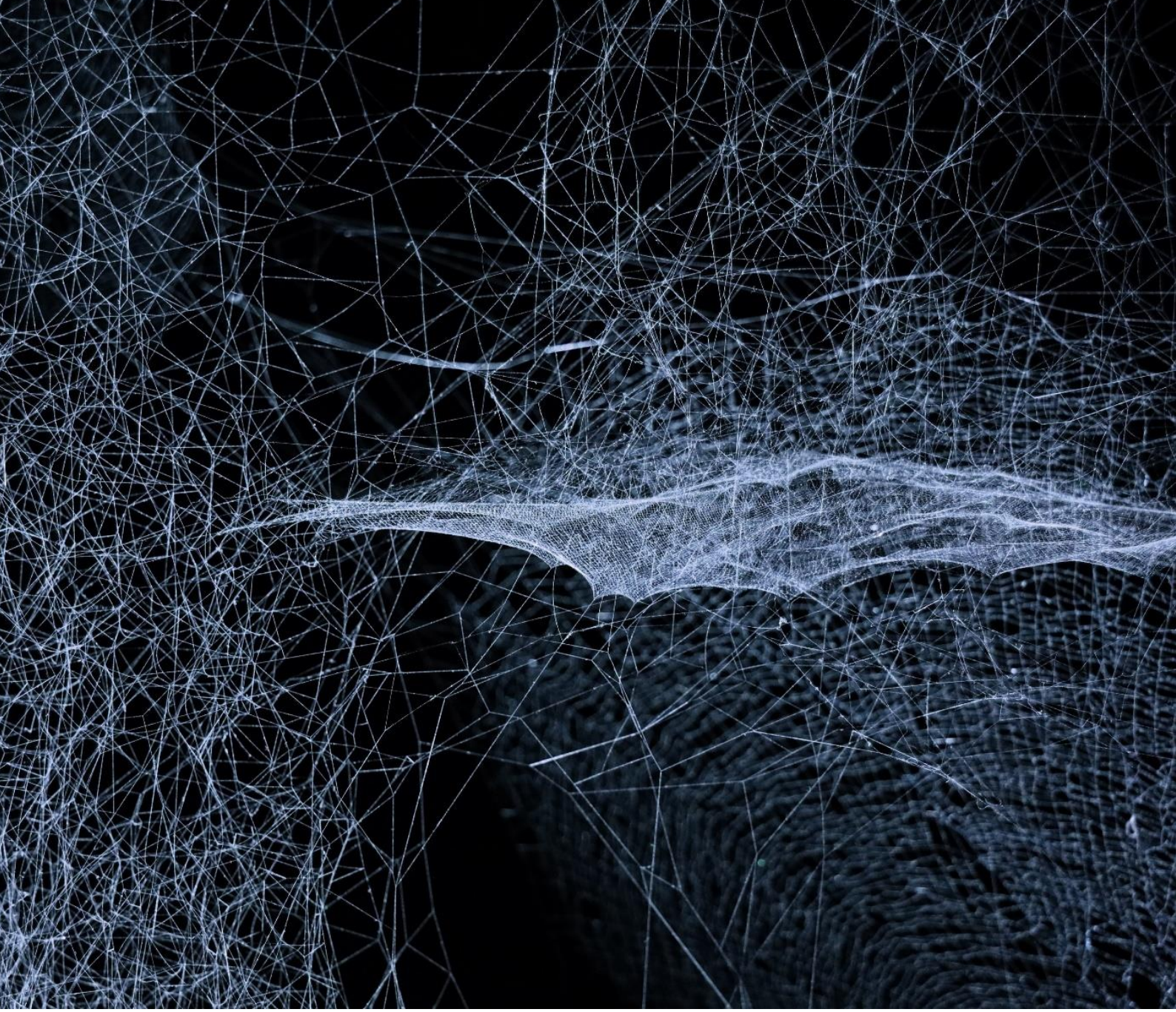
If your system can flag content that has been auto-tagged, this might be all you need. A metadata editor finds the auto-tagged content, checks the tags for accuracy, makes changes as needed, and marks the record “verified,” or some other status that lets users know they can rely on the metadata therein.

Another time saver can be using a tagging service to subgroup content that a human subject matter expert (SME) can then describe further. For example, while the tagging service might not be able to identify specific product variations, it can likely tell the difference between your

tractors and your lawn mowers. By doing some high-level categorization, you can assign each subgroup to the appropriate product manager or other SME for further details.

As a rule, the more generic your content or tagging requirements are, the more value you can derive from an auto-tagging service. In addition, the larger your collections, the more enticing such a service might be.

But if your content system manages medical content for veterinarians, “puppy with red ball” won’t likely be a suitable tag to describe an image intended to illustrate a breed, stage of life or visible signs of disease.



Creating semantic links

Computer-based search operations have historically been based on character matching: a search for *building* finds content tagged with *building*.

One limitation of this approach is that the search engine has no real understanding of the user's intentions or expectations. For example, was the user thinking of *building* in the verb or

the noun sense? Was she looking for content related to physical structures or was she looking for content showing a child constructing a fort?

In order to provide meaningful search results, the search engine must have a semantic understanding of the user's meaning. There are a few ways in which this can happen:

- The user can provide additional terms in the search
- The search engine can consider previous searches by the user to get a sense of intention
- The search engine can consider public trend or popularity to assume one over the other

Google uses additional factors, such as the searching user's location and local time of search, to establish semantic understanding. For example, if you type *pizza* into Google around lunchtime, the engine's assumption is that you want one, not that you want to find out how many are sold each year, or learn to make your own. Based on your location at the time of the search, Google prioritizes results near you.

You can influence Google's semantic understanding of your search by providing additional terms, such as "how to make pizza" or "origin of pizza." With these few additional terms, you "educate" Google about your intentions and expectations.

Some content systems enable users to make semantic connections between content, between content and terms (tags), or between terms themselves. Search engines can consider these links in making search suggestions, "more articles like this" suggestions or similar guidance.

There are two factors involved with making this possible in a content system:

1. What options are there for users to make semantic connections?
2. How can the system's search engine use these relationships to guide users?

Say, for example, you had photos that were taken at an event in Berlin. Unique to those photos might be the event name and date. But when it comes to location, perhaps you assign the tag *Berlin* from a list of cities that has already been established.

Now, assume that *Berlin* tag has its own tag descriptors, such as *Germany* (the country) and *German* (the prevailing local language, chosen from a list of known languages). By extension, say that the *Germany* tag has a tag descriptor for Europe.

Without specifically assigning *German*, *Germany* or *Europe* to the event photo, those terms are still accessible to search operations. For example, "events in Europe" might find the Berlin event photos; likewise, "events where German is spoken" might also find them.

Again, what is important to remember about the value of semantics in content systems is that, in addition to making it possible for users to define relationships, the search engine must be "smart" enough to consider the user's search criteria in the context of all available terms.

"Events where German is spoken" requires that the search engine understand that, in this context, *German* would be a language. But German is also a nationality and a descriptor of

culture, among other things. *German Architecture*, for example, doesn't describe buildings that speak German or carry German passports.

Semantics provides the machine with context. When you ask a child, "what did you do in school today?" you are not looking to hear, "I walked and listened and looked and played and learned." You have a clear idea of the kind of response you are expecting and—sarcasm and misbehavior aside—so does your child.

As humans, we are natively aware of semantics in communication. When semantic understanding is missing in a conversation, we ask for clarification. Machines must derive clarification in other ways.

Archiving content

It is a good idea to think of archiving as nothing more than a change in lifecycle stage for content. Archived content might be reusable at any time, so the notion that you are “done” with any content forever might result in missed opportunities.

The act of archiving should be directed by policy: What gets archived, why and when?

Some of your content might never be subject to archiving. For example, your corporate logo and stock photography collections are initially assumed to provide value indefinitely. But if at

some point, you design a new logo or decide that some stock content is no longer suitable, you can manually archive what you no longer need.

Other content should have archiving rules defined in the policy definition of the content. An annual report might be slated for archiving a year from publication, or at the publication of the successor report, whichever comes first. A product brochure might be slated for archiving when the product is updated or discontinued. Campaign materials might be archived at the end of a campaign or never, depending on whether you expect to reuse them.

Your content system should be configured to adhere to these policies. For extra peace of mind, perhaps you have the system notify someone before content for which he is responsible is archived.

What “archive” actually means should also be defined by policy. At some organizations, it is nothing more than a status change from “Available” to “Archived.” The content remains searchable and accessible to permitted users; only the status has changed. Other organizations opt to move archived content to alternate storage media, or take it offline entirely. Some content systems can move content automatically, so the process of archiving is completely transparent to users.

As a rule, it is a good idea to keep archived content “findable” in search operations, even if those operations are limited to only a select group of users. There are many reasons for which archived content might need to be found:

- Reports that reference older content, or indicate when older content is still in use
- Research into older campaign performance
- Litigation in which archived content can support the prosecution or defense
- “Flashback” graphics or corporate heritage reports
- Design reference for consistency or contrast

A point to remember with regard to archiving is that today’s trash can become tomorrow’s treasure. A television commercial that shows flight attendants with outdated uniforms and hairstyles might seem stale when only a few years old; but 25 years later, that same footage could become a viral Internet sensation.

Content Routing

When we speak of routing content, we refer to making content available in additional or different places. In some cases, this involves actually moving content, but in many instances it refers only to changing access permissions, sharing content or links, or opening portals. The goal is to make content available anywhere it is needed, in the formats that are needed.

Most important among the concerns for routing content are:

- Routing content vs. routing metadata
- Making content available to users and the public
- Routing content between business systems
- Publishing content to output channels

Collectively, these considerations account for the vast majority of content routing requirements.

Routing content vs. routing metadata

In the traditional digital asset management paradigm, there is a distinct difference between content and metadata: the file is the content and the DAM system (database) record contains the metadata that describes the contents of that file.

For file-based content, this paradigm still works. But “fileless” content is increasingly common. Emails, blog posts, tweets and Google Docs are common examples.

To be technically correct, these content types also have associated files, such as the file of the database where the Google doc is stored. What is different is that the user is not exposed to the underlying files for these and other content types. Still, content needs to be managed, no matter what the format.

Another type of fileless content is content that is created as a result of combining other content elements into a single entity.

Consider any given website page. The page text might have come from a Word document; the images from individual files created in Photoshop; and the videos linked to from YouTube. But how would you manage the page as a whole? How would you add metadata to make the page findable from within a content system, or more readily accessible for research. Where would you flag, for example, that the page includes a reference that will need updating in 6 months?

Most website content management systems (WCMS) lack flexible metadata options. They might permit a tag or two, but little more. A content system can be used to manage website pages, even if the pages themselves are hosted from another system.

With that website page in mind, think about a campaign landing page or a press kit. In these cases, you also have a collection of content serving as a single piece of content. What is different here is that you *could* create content like this in a content system. Using templates, as previously discussed, new content of these and other types could be created fairly easily. Then, of course, they would also need their own metadata profiles to define and manage them.

Where fileless content is considered, metadata is the asset. When that metadata describes a remote page or video, it enables users to find those resources without having to know what to search for on the Internet. And when that metadata is itself the content, as is the case when building landing pages or other such templated-based content within the content system, the metadata is all there is.

Metadata without Data

The term *metadata* suggests there is *data*. This distinction becomes confusing when the data is the metadata, and vice versa. For the purposes of the next generation of content management, assume that the line between the two will continue to blur.

So, when it comes to routing content, this refers to either an underlying content file *or* (meta)data that is itself the content.

Making content available

If your content system will include access for human users, you will likely want to provide some means for routing content to humans.

Content becomes available to your known users through permissions that enable them to access the content. Content can then be shared by permitted users to others outside your system through shares. (Assuming you have enabled such functionality.)

By changing the permissions that affect a given user's account, you can change what content that user sees and accesses. (Optionally, of course, you can disable the account if you want the user to have no further access.)

Shares that are created by users can include access controls. Expiration dates, and whether to enable downloads and re-shares are common options. These shares can also be embedded into websites and other locations where you want the content displayed.

Another good way to make content accessible to those outside your content system is through the use of portals or microsites. In practice, there is little difference between the two, as far as users are concerned. In both cases, the user connects to a website location where content that resides in the content system is accessible, often without having to log in to an account.

Depending on system capabilities and configurations, users can search and browse for content, preview content, see metadata, download and share.

In some cases, the portal is a standalone site, meaning it has its own URL and, when loaded, the portal is all the user sees. In other cases, the portal can be embedded into an existing website page. This option is more popular when user interactivity does not require a fully featured interface, or when only a few pieces of content will be made available.

For example, your content system might be used to manage a million different pieces of content, but when users connect to your product documentation page, you want them to see only product documentation. In limiting what users see, you can improve their experience by preventing them from searching for what they need.

Routing content between business systems

An API (application programming interface) enables software developers to create add-ons to extend system functionality, build connectors to move data between systems, design alternate user interfaces, and more.

A content system designed “API first” was built with the expectation that it would be connected to other systems. By contrast, some systems have APIs that were later added to the core product or, in some cases, there is no API at all.

The Meaning of API First

“API first” does not mean a system has no user interface. It means only that API accessibility is considered a primary means for working with the system, so the API must enable remote systems to perform reasonable workflows, actions or processes within the system.

All APIs are limited in what they can do. In theory, an API enables a machine user to do in the system whatever a human user can do. In some cases, an API even provides even more functionality to machine users than to humans. For example, while a human user might see a menu option to create *one* new file, an API might enable a remote system to create a thousand new files in a single API call.

A well designed API considers the different use cases between human and machine users, and limits or extends available functionality accordingly. This is why an API-first system can be expected to do more in a multi system environment.

Content systems can be very valuable when used as hubs between a number of content-consuming business systems. In this capacity, they serve a number of important advantages:

- Content is centralized, accessible to all systems that need it
- Developers can rely on a single central system when considering content routing needs

That last point is important because a centralized content hub provides developers with a constant when connecting systems. In other words, instead of trying to figure out how to best connect two systems to one another directly, all they need to know is how to connect each of those systems to the content hub—a system that is already known. This removes a variable in each instance, which can lead to more reliable and effective results.

API Quality and Completeness

When evaluating content system components, do so with the assistance of a developer or other technical expert who can accurately assess the value of the component's API. Some APIs offer only limited functionality or performance that might make them unsuitable for your plans and goals. In addition, it is important that sample code be made available in the programming languages your team uses. These software developer kits (SDK) can be the difference between your developers becoming productive immediately, or spending weeks researching how things should be done.

In addition, when each business system is connected to the central hub, this greatly reduces the numbers of system connections that are required. Consider having just three different content systems, one of which is the hub. In order for all three systems to share content, only two connections are required. But if the connections are system to system, three are required.

The math becomes more complex as more systems are added. For example, interconnecting five content systems without a hub would require some 10 different connections. Using a hub, that number is reduced to four. Not only does this save significant financial, development and time resources, it becomes far easier to maintain over time. A network of too many system to system connections can be more prone to technical issues, and far more complex to debug when something does go wrong.

Then, with all relevant systems connected, you gain the benefit of being able to route content between them. Reuse of content is common today, but virtually all content consuming systems treat content as a protected asset, unavailable outside that system. This is common with website content management systems (WCMS), marketing automation systems (MA), product information management systems (PIM) and more.

By contrast, when systems are connected via API, they can share information. This provides two important benefits:

- Users are not required to add or update content in more than one system
- A single system can serve as the master for content, providing a "single source of truth" that helps ensure accuracy and consistency

The concept of a master source of information is encouraged in information management theory, but this is not always the practice in multi-system software environments. The trouble is that, with each business system considering itself the master of all content it contains, users are forced to feed information to multiple "masters," or arbitrarily choose one system to be the master of one type of data, while another serves as master for another type.

The most common solution in a multisystem environment is to grant the role of “master” to the system most closely associated with the information.

For example, product information is managed in the product information management system, which is recognized as the master source for that information. Marketing campaigns are managed in the marketing automation system, which serves as the master for that information. And the customer relationship management system (CRM) serves as the master for customer and prospect information.

Anyone who has connected a marketing automation system (MA) to a CRM knows the complexities involved. It makes some sense that contact information be managed in the CRM. After all, this is the point of that system. But the MA also uses that information. This means the data must be sent from the CRM to the MA, and updated as needed. But what happens when a customer updates her contact information based on an email that was sent from the MA? Logic dictates that the MA send the update to the CRM.

But what happens if both customer records have been edited since they were last synchronized? For example, the MA might have received a new mailing address, but a new contact phone was entered into the CRM. Both records are now updated, without the other system being informed. When the synchronization finally occurs, the system is left to decide which data is most recent.

More sophisticated integrations will be able to update records on a per field basis, which solves this problem in most cases. But this “simple” program involves only two systems. When you add other systems to the mix, such as a system that handles purchase information for that customer, or another that manages support requests, things become very complicated.

Before long, it becomes unclear to both machine and human users which system is the master of any given data, or even which data is most current or accurate. If you have merged records in a CRM, you might have experienced that hesitation while you considered which record held the most recent or complete updates of each data field.

The concept of master data management (MDM) refers to having a plan in place that describes, without ambiguity, the sources from where information comes, which system owns the information, and how information is shared to “downstream” systems that need it.

In an ideal world, a single system would serve as the master of all business content, be that customer or product information, marketing messaging and campaign results, technical documentation, contracts and orders, and everything else. But the diverse use cases for these content types suggest that there will not likely be such a system coming any time soon.

A next-best solution would be to abstract the content that is common to all systems so that it can be managed in a single system that serves as the primary content router between systems. From there, the content can be filtered down to the more specific systems from where additional information can be added and, perhaps, flow further downstream to additional systems or output channels.

In a way, you can think of this as a content pyramid, with the most common content at the top, and the more granular content below. Consider these examples of content that is of use across all systems, and should be consistent across all systems:

- Copyrights, disclaimers or other legal notices
- Contact data, such as emails, addresses and phone numbers
- Official logos, executive headshots, or other such graphical content

Content like this needs to be managed by persons authorized to make changes, which won't likely include the majority of your workforce. Further, that official and sanctioned content needs to be readily accessible when it is needed.

Without the use of a master system, copies of this content would be added to each system. When the source content is updated, each location where that content has been copied would have to likewise be updated. The problem is that this rarely happens. The result is that materials get released that include old logos, outdated office addresses, or worse.

One of the original promises of Digital Asset Management was that content like this could be stored in a central location from where people would get it when they needed it. The problem with this approach is that it does not alleviate the primary problem, which was that copies of the content needed to be added to other systems.

When systems are configured to pull content, on demand, from a master source, the chances of outdated content getting released are greatly reduced. For example, if your copyright notice is embedded as a link into templates or other locations where it is used, the users creating content with those template would not have to worry about whether the notice was accurate. Knowing it was coming from the official source, the user could rest assured that what would be published was accurate.

This concept is especially important when the source content is more complex, such as a end user license agreements (EULA) or terms of use. When a software developer is building an install wizard, she will not read the EULA to determine if it is current. In fact, she will likely have no idea what the software's terms of use are—that is the job of the Legal department. All she wants is a link from where she can pull the most recent version of the EULA and include it in her build.

This is great example of more granular content into which more general content is embedded. It makes no sense to store software source code in a system not designed to manage source code, considering the unique requirements of software development. But it also makes no sense to manage copyright notices, disclaimers, EULAs and other content that will be included in software builds in systems where they cannot be reliably updated.

When considering all the business systems in use at your organization, consider what unique content is created with each, and identify the content that should be shared across all systems. This will help you illustrate a pyramid that defines the most reasonable master for each content type. And this, in turn, will enable you to plan the best ways in which to connect your systems.

Publishing content to output channels

Though output channels can be as diverse as Twitter, a video distribution service, or product catalog production, what is common when routing content to output channels is the workflow considerations through which content must traverse before it is published.

Unlike moving content between *internal* business systems, moving content from, say, your PIM to your Facebook account, can move markets, perhaps unintentionally.

Before getting into the content considerations of publishing to channels, consider some of the more common output channels:

- Social media (Twitter, Instagram, Facebook, etc.)
- Video distribution (Brightcove, YouTube, Vimeo, Kaltura, etc.)
- Microsites (custom developed additions to corporate or partner website properties)
- Printed production services (books, catalogs, brochures, fliers, etc.)

The requirements and considerations of these diverse channels can greatly affect how content should be managed before it is published through them.

An extreme and obvious example is that an announcement crafted for Twitter is not suitable for publication into book form. But there are many less obvious examples that should be governed by policy and common sense.

The goal is to ensure that what gets published is:

- Complete and correct
- Suitable for the chosen output channel
- Approved for publication by an authorized source
- Ready for publication

Content is complete and correct

At the core of all content publishing is making sure the content is complete and correct. As a concept, this is pretty fundamental. The concern is when you have hundreds of people making publishing decisions, and you have millions of pieces of content, it is important that “complete and correct” be clearly indicated in the content system.

Ideally, statuses like this should be so obvious that anyone can see them, without having to know where to look or what to look for. There are certain things you do not want to rely on user training to make known, and publication status is one of those things.

Metadata controls make changing status easy enough, but also make sure that the status itself is clearly visible to anyone who views the content record. It is not enough for a small field to be set to “Do Not Publish,” if that field is not obvious.

In most cases, a status like this will require approval from a content authority. Keep in mind, this is not necessarily the same authority who will approve the content for publication. At this point, your concern is only whether the content is correct and complete, even if it should not be published for another year.

Content is suitable for the chosen output channel

The next consideration is whether the content is suitable for the chosen output channel. Obvious considerations are that you do not send a spreadsheet to YouTube. But even within suitable content types there can be unsuitable content.

Image formats are one such consideration: After a designer creates the movie poster for your next blockbuster film, there will likely be a very large Photoshop master file. This is not the

content version you would share to social media. Instead, you would share a (much) smaller PNG or JPEG version of the master. If your content system can create this for you, all the better; if not, someone will need to create it manually, or an upstream integrated system must create it.

Even when a given format could be published via the selected channel, you might not want that to happen.

For example, when you publish a PowerPoint slide deck, you provide an editable file. Anyone can download that file, edit it, and perhaps upload or share it again. You might like what they added, but you might not. Instead, publishing a slide deck as a PDF file can ensure the content is distributed in a non-editable or less easily edited format. No matter where it is reshared, you will be more certain that is it the content you originally published.

Define within your content management policies which content formats are suitable for which output channel you will use. If a suitable format is unavailable, make clear to the user how to inquire about getting the content in a suitable format. This might be as simple as a directive in the content record that reads, “To get this content in other formats, contact …”

To improve the user experience further, make clear in your content systems which formats are available. Even better, provide an “easy” sharing option through which users can think in terms of “I want to share this to Facebook” rather than “I need this in JPG format.”

Content has been approved for publication by an authorized source

Unlike a content editor/approver, who might look for typos or other mistakes, some authorized source must conclude that a given piece of content *may* be published. A given tweet might be less than 140 characters and contain no errors, but that does not mean the content of the tweet should be released.

In some cases, the content approver will be the same person as the publication authority; but this should be clearly defined for each content type.

Further, your content system should make publication impossible until the required approvals have come in. This can be made possible by providing a metadata field (such as a checkbox) that is accessible only to authorized approvers. When the box is checked, the system and its users know the approval was real. If the box is unchecked, the system knows to not route the content to any output channel.

Improve the user experience by making it clear to users why a given piece of content cannot be published. This will help them understand the rules, and it will reduce support requests asking why they cannot publish something. Consider shielding sensitive, non-approved content from such users via permissions.

Keep in mind that resourceful users might discover that they can download and publish content manually, even when the “publish” controls in the content system do not permit them to do so. This would possible when the user has download rights even for content that is not approved for publication. This is a very likely scenario. If users do not understand the policy, they might think they are merely overcoming a glitch in the system by circumventing it.

Content is ready for publication

Content can be complete and correct, available in a suitable format, approved for publication and still not be *ready* for publication. A press release that is under embargo is one example; next week's quarterly results report is another.

By making "ready for publication" its own attribute, you can save approval authorities from withholding approvals. For example, someone charged with approving that quarterly report might fear that if she approves it, it will go out too soon. Instead, she will make a note for herself to approve it next week, and then be out sick on the day it is needed.

But if the "ready for publication" policy is clear, and clearly indicated on the content record, such as showing an embargo date, the approver knows that her approval is conditional upon all required considerations being met.

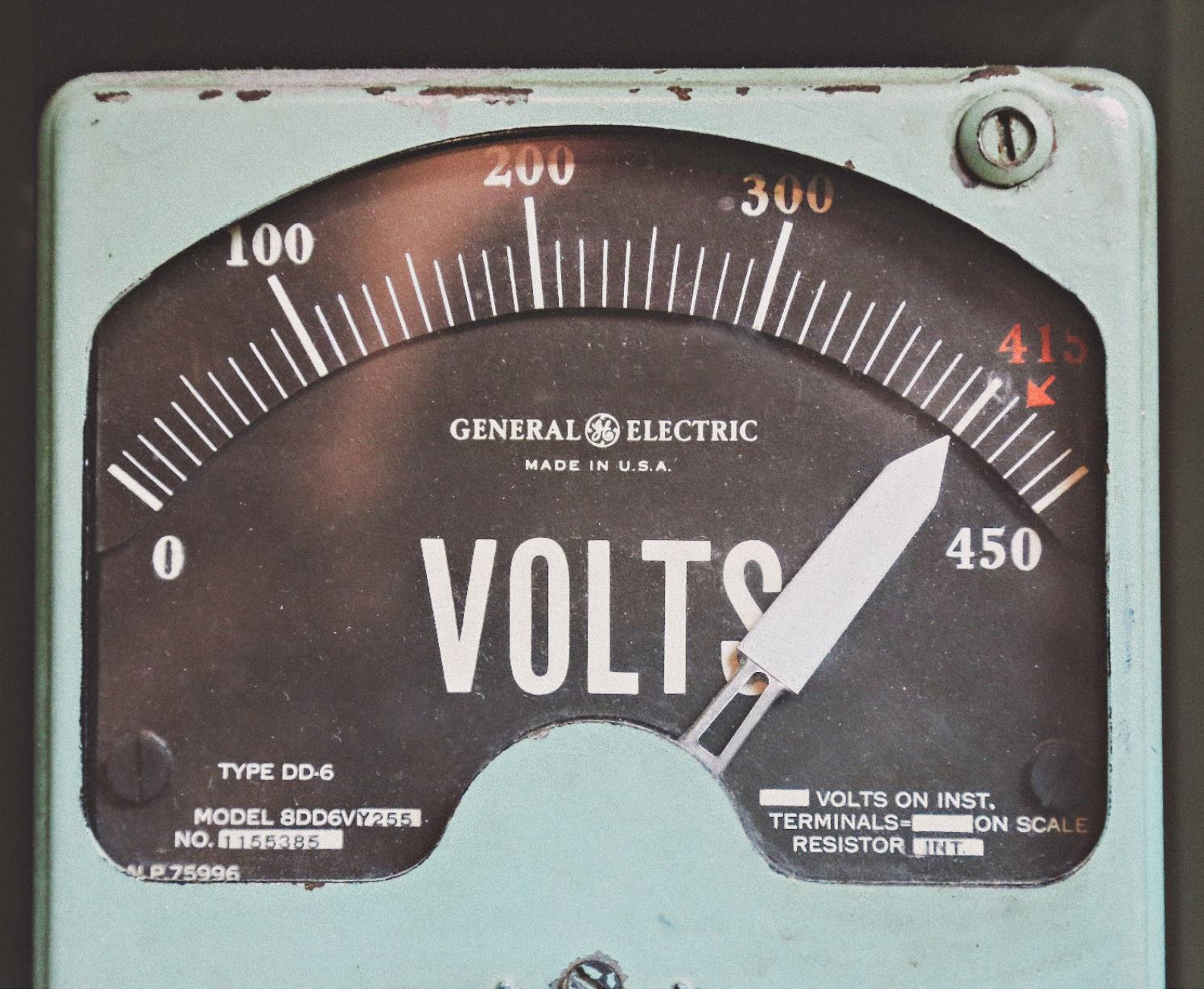
Considerations like "ready" are typically rules based. While the indication can be communicated via a metadata field, as is the case with the approval, the authority of the "ready" status might be a rule that the content system can evaluate.

For example:

The content must be marked complete and correct by a user in the Editor role AND
The content must be available in a format suitable for publication AND
The content must be approved by a user in the Manager role AND
The Embargo Date field must be empty or include a date in the past.

When all these conditions are met, the content system marks the content "ready for publication."

You will likely have different rules that apply to different content types. This is yet another reason why thinking of content in terms of content types make content management easier.



Measuring Results

There are only two good reasons for measuring the results of your content system:

- You need to report those results
- You plan to use those results to improve performance

The first point suggests that some manager will qualify the value of the content system or the content based on usage and other statistics. The second point suggests that a system manager will make changes in order to improve future results.

You might have specific things that you need to know about your system, but there are a few measurements that are common:

- What content and system features are getting used
- Are users happy with content collections and system functionality
- Do other system components need to be brought into the fold
- Is the system sustainable and affordable

What content and system features are getting used

Knowing what content in your system is most widely used enables you to make a few decisions:

- What types of content are most useful
- What types of content warrant acquisition investment changes
- Which content types should be archived or stored on faster, more accessible storage (e.g. CDN)

Statistics Hoarding

Before you decide that measurement is a requirement for you, and before you use this as a consideration about which content system components will work for you, understand why you want those results, and how they will be used. Tracking data for the sake of data collection is pointless, if that data will never be considered.

You might have additional interests, such as featuring "this week's most popular content," or rewarding your more popular content authors.

In order to make these determinations, there are few basics that you'll want to know:

- Which pieces of content are getting downloaded most
- Which pieces of content are getting shared most
- When content is shared via private links, is the recipient opening it
- Is content being reshared

It will be up to a component of your greater content system to collect these numbers, but you might find an external analytics engine more helpful in helping you make decisions about the data.

For example, most website content management systems offer some means for tracking popular blog posts and such; but virtually all marketing teams rely on a external tool like Google Analytics to present a more meaningful picture of the data. Likewise, you might find such a tool to be valuable in crunching the numbers associated with your content access and use.

An additional benefit to using an external tool is that the data can be immediately accessible to those who might have no other interest in your content system. For example, IT teams that need to make decisions about storage purchases might like to see that data, even if they have no need to access your content repositories.

Are users happy with content collections and system functionality

One thing you cannot track with usage statistics or downloads is what content users did *not* find. Likewise, while they might use a given menu option most, this does not necessarily mean they find that option to be convenient.

It is important to remember that usage statistics show *usage*; they do not necessarily show appreciation.

There are a few ways in which to measure user appreciation of your content collections and the system itself:

- Instant feedback mechanisms
- Surveys
- Interviews

Instant, easy feedback options are valuable because users will not bother to report a problem if it is not convenient to do so. Provide options in your system for users to report problems, ask questions or provide feedback.

Examples include:

- On content records: Do you like this content?
- In search results: Did you find what you expected?
- In user settings: Do you need help with configuring your account?

Context-relevant options will be welcomed by users, and they can reduce your human-provided support requirements.

Surveys are attractive options to those who seek the information, but they are not generally popular with users. Some users will fill out a survey, but this subset might not represent a meaningful cross section of your user base.

If you do conduct surveys, here are a few suggestions:

- Keep them short and focused
- Keep questions simple and unambiguous
- Structure questions so that answers are measureable
- Do not send surveys too frequently
- Reward participation, e.g. by sharing the results

Shorter surveys seem less daunting to users. Users are more likely to abandon a survey when they load it and see a “next page” option.

Do other system components need to be brought into the fold

Based on user input and actions, you might decide that additional components need to be added to your system. This could be as simple as additional computing resources that are needed to handle loads, or it could mean additional functionality that users request or require provisioned via smaller software components you integrated, or complete business systems to which you connect.

One popular promise of Cloud Computing is that additional resources can be provided on demand, and released when no longer necessary. While this is technically correct in some hosting centers, there is additional expense involved with provisioning added horsepower, and

even scaling down might at times result in paying more if you're leaving a discount tier, so it is not something likely to happen without your consent.

If your content system is run on-premise, the additional compute resources are up to you to provide. If you find that a significant investment in hardware is required, you might speak to a systems consultant to see if some of your system components can be transferred into the cloud, thereby freeing up local resources. This will have to make sense technologically, and you will have to make sure that in doing so you do not violate any policy or other regulations that might affect your content or the management of other data the system processes.

Inherited Content and Systems via Acquisitions

One unexpected surprise to your content planning could be the acquisition of another company's collections and management systems. Despite all your careful planning, you might find you need to one day make room for collections that have not been so carefully planned, or that might be managed on inaccessible systems. In some cases, it might be worth letting content remain in an external system that is connected via API rather than "dumping" unmanaged content into your system. In time, you can plan for a more controlled means for incorporating the acquired content. But always be careful to not jeopardize the quality of your system by exposing it to content for which there is no plan.

Other system additions could be additional business systems that must be connected to the greater content system, which could be existing systems or entirely new systems, like a dedicated approval workflow or creative collaboration system. In making these decisions, consider the same factors you considered when designing the core system:

- What content will the new system process?
- Of what, if any, content will the new system become the master source?
- What will be the system-to-system connectivity requirements?
- Does the proposed new system offer an API suitable for it to be connected to the existing system?

One reality of large enterprise software systems is that customers rarely use the entirety of a given system's capabilities. For this reason, when a new functionality requirement presents itself, determine if an existing system component can perform the new task before you invest in an entirely new system.

Is the system sustainable and affordable

The final measurement to determine, no matter what other measurements are important to you, is whether your system is sustainable and affordable. A system that is running great and making a large number of users happy might, in fact, not be sustainable for one or more of the following reasons:

- The system is too costly to maintain
- The system requires maintenance resources that are not available
- A known future business requirement will obsolete one or more parts of the system

When purchasing new enterprise software, organizations often look to startup costs and annual maintenance. If both figures are in line, the system is deemed affordable.

The trouble can arise if the scope of the initial system proves inadequate. For example, say that the required hardware computing resources was underestimated, or that users presented a functionality requirement that was not considered.

While you might successfully scramble to come up with the purchase price of whatever is needed, you must also factor those system additions into the ongoing maintenance.

Be mindful of custom development, too. While a flexible system can be customized to do exactly what you need, it is important to know how those customizations will affect future costs. For example, if core component updates will each require additional coding and testing to work properly, the ongoing benefits of custom functionality might not be worth the expense.

An additional consideration to ongoing sustainability for a content system is that there must be one or more people who manage it. It is always great to have a system “champion,” who promotes and defends the system, wherever possible.

But the ultimate quality and usefulness of the system will depend on folks doing a good job of keeping the content and metadata clean and those enabling these contents to be brought were consumed. The value of content is zero (or even negative) if it is not accessible and made available where needed.

Without such attention, a system that is considered a winner this year, might become less useful next year, and virtually useless in the years after that.

The introduction of an enterprise wide content system requires discussion, planning, knowledge, resources, policy, good timing and a bit of luck too. If you cannot dedicate reasonable time and resources to the research you will need to make smarter decisions, consider whether you can afford for the effort to fail.

In most cases, content systems that are not adequately designed, funded, managed and maintained do fail. At the very least, it becomes a bad investment for the organization. In worse cases, it can adversely affect other aspects of the business.

But when all the pieces are in place, and there is a plan for ongoing support and maintenance of the system, ready access to content can greatly improve the value that individuals and departments can provide.

Next Steps

This document has provided you with one perspective on best practices for managing content. And while the recommendations herein are based on practical experiences, there are a number of differing opinions on some aspects of Content Management.

It is a good idea to read further on the subject, and to speak to others who have already built and managed similar systems. The following resources are recommended.

Content Management Books

Metadata for Content Management - Designing taxonomy, metadata, policy and workflow to make digital content systems better for users

Metadata for Content Management helps digital content managers design better content organization strategies, and envision and deploy creative ways in which metadata, taxonomy, policy and workflow can be used to make digital content systems more usable, functional and valuable to users. Available on Amazon

DAM Survival Guide - Digital Asset Management Initiative Planning

DAM Survival Guide is a digital asset management book that explains everything you need to know to design, plan, deploy, promote and maintain a successful DAM initiative at your organization. Written by a recognized and awarded DAM industry expert in a friendly, easy-to-follow style, DAM Survival Guide is a must-have resource for those new to DAM, and it's great for those looking to increase their DAM knowledge too. Available on Amazon

Free from Picturepark

Picturepark has a limited number of the previous two titles on hand. Contact info@picturepark.com if you are interested in receiving a copy of either title.

Digital and Marketing Asset Management: The Real Story About DAM Technology and Practices

Digital and Marketing Asset Management was written for both marketers and publishers who need a clear understanding of how technology can empower them to do their jobs better. For technologists, the book is also a technical dissection of how DAM really works, so you'll have a clearer vision of how DAM could and should fit into your enterprise architecture. Available on Amazon

DAM Guru Program

DAM Guru Program is an online community that includes more than 1,000 members from around the world. Experts in the fields of digital asset management, content management, information science and information technology offer their help to those who need it. All member services are free of charge.

Credits & Copyright

Author

David Diamond

Photo credits

[1] Photo by Barn Images on Unsplash (Overview)
[2] Photo by kazuend on Unsplash (User and Flow)
[3] Photo by Todd Diemer on Unsplash (User Groups and Roles)
[4] Photo by MJ S on Unsplash (Content Creation and Acquisition)
[5] Photo by Cristina Gottardi on Unsplash (Content Management and Collaboration)
[6] Photo by Eugenio Mazzone on Unsplash (Storage and Archiving)
[7] Photo by Eric Rothermel on Unsplash (Collaborative Communication)
[8] Photo by Shane Hauser on Unsplash Adding Real-World Metadata)
[9] Photo by David Kovalenko on Unsplash (Validating and Improvig Automated Metadata)
[10] Unknown
[11] Photo by Samuel Zeller on Unsplash (Archiving Content)
[12] Photo by Sam Poullain on Unsplash (Content Routing)
[13] Photo by Rob Laughter on Unsplash (Making Content Available to Users and Public)
[14] Photo by G. Crescoli on Unsplash (Publishing Content to Output Channels)
[15] Photo by Thomas Kelley on Unsplash (Measuring Results)

Copyright & Licenses