



7 Features You Need From Your CMS

And other tips to help you choose
the right platform



Foreword

If you are preparing for a digital transformation or are already in the midst of one, you have probably realized the content management system (CMS) you choose is a critical part of the process. The right CMS can help your organization deliver memorable customer experiences on the web and beyond. But if you pick technology that doesn't truly suit your needs, your entire organization could quickly find the CMS is a blocker to success, not a catalyst for growth and innovation.

This eBook is for those of you who are ready to consider a new CMS or just want to learn more about ways to manage your content more effectively. In this document, we break down seven key features for your next CMS and discuss some other important concepts such as web frameworks, workflow management and responsive vs adaptive design. Your organization likely has additional requirements not covered here but we feel these are some of the most important things to consider when you evaluate the content management software on the market today.

Table of Contents

| | |
|---------------------------------------|----|
| A Flexible Repository | 4 |
| Open APIs | 5 |
| Granular User Permissions | 6 |
| Multilingual Content Management | 7 |
| Multisite Support | 9 |
| A Powerful Web Template System | 10 |
| Web Content Personalization | 11 |

1 A Flexible Repository

Web development is no longer just about websites. In today's multi-screen world, the content we create can be consumed by a wide range of devices and platforms, including ones we never imagined would be part of our customer experience strategy such as smart TVs, digital kiosks and other connected devices. But the reality is, the days of single-channel user engagement are long gone, and if your business is going to thrive, you need to think and act beyond the web.

A CMS that gives structure to content is critical today, and this begins with the repository, which stores your content—articles, images, videos, audio files, and so on—in a central, accessible location. You should ensure the repository gives you the flexibility to structure your content, customize the content model and make the best use—and reuse—of your digital assets.

Moving from a page- and template-centric WYSIWYG approach (what you see is what you get) to a more structured content-model approach allows you to efficiently deliver specific pieces of content that can be consumed independently, or combined to create a page.

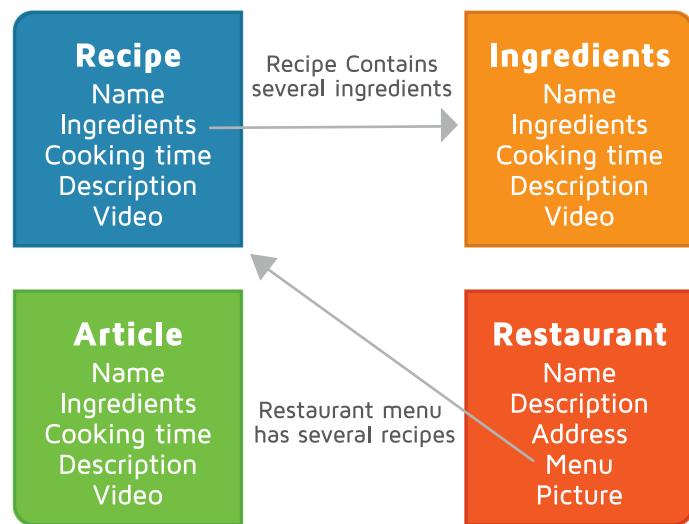
We can't assume the design applied to content will be appropriate for every device or channel, so look for a CMS with a strong separation between content and presentation. This allows editors to focus on the content itself rather than layout and design.

What is structured content?

Structured content means content is separated into independent parts. The independent parts are often labeled "fields" in the context of a CMS but are commonly called "chunks" by content strategists. Because these parts are separate entities, they can be edited independently, and used in different contexts. Fields for a blog post may include: blog post title, author name, author job title, author organization, blog post description and so on.

By structuring your content, you can take advantage of a CMS to draw semantic relationships between pieces of content, reuse your content and much more. Take recipes, for example. You can link

recipes to ingredients, restaurants, allergens, cooking times and other fields. In turn, you can easily provide users a list of recipes that contain a specific ingredient or ones that have a certain cooking time. On the other hand, structuring your content puts your users in control, and it becomes much easier for them to find what they are looking for on your site or app, and even discover information they otherwise would never have known.



Creating semantic links between content objects can have immense impact on the user experience and ultimately your digital success.

Key benefits of structured content

- Make your content more discoverable by providing users with related content easily and implement a taxonomy-based search.
- Engage users everywhere by feeding content to multiple devices, channels and platforms automatically.
- Take full advantage of your content by combining various pieces into new bundles or collections.
- Personalize the user experience by delivering dynamic content to a target audience.

[Learn more on the eZ Blog >>](#)

2 Open APIs

Usage of mobile devices continues to grow fast and content consumption is no longer limited to the home or office. According to Pew Research, 64% of Americans owned a smartphone in 2015, up from 35% in the spring of 2011. Consumers are on the go and they expect to be able to engage with brands anytime and anywhere.

Devices connected to the Internet are also gaining popularity and smart watches, smart appliances, and smart cars are only a few examples of the Internet of Things (IoT) that are now a reality. By 2020, Gartner believes tens of billions of everyday devices will be equipped with sensors to process and analyze data. Our refrigerators will tell us when our food spoils, our cars will drive us to work, and our smart watch will warn us of impending health problems.

What does this have to do with your CMS? Well, today's modern CMS can help you go above and beyond building websites, and help you engage users on different devices, channels and platforms. You want to look for a CMS with a flexible repository—remember, structured content is key—and open APIs. With this combination your CMS can deliver content beyond the web—even the smart refrigerator that orders you a fresh dozen when your eggs are about to go bad. This can mean real business value here and now, and years into the future.

Let's take the recipes example again. You could create an iPhone app for people with gluten sensitivity and deliver the right recipes from your CMS via what's commonly called Content as a Service (CaaS). You would simply point your app developer to the CMS' API and fetch recipes that do not include gluten, and voila, you have a targeted app using existing content that could potentially deliver real business value. Of course, you could also repurpose the content in cooking books, on websites, mobile, native apps, Facebook Messenger and other platforms.

Integrating your CMS with other systems

If you'd like to engage your audience and introduce business efficiencies, it's crucial to integrate your CMS with your internal and external systems.

For example, e-Commerce companies may want to integrate their CMS with their CRM, ERP and payment processing systems and publishers may want to syndicate their content to third-party platforms such as Apple News and Facebook Instant Articles. You should ensure your CMS has open APIs to facilitate a connection to other platforms and systems.

Be sure to ask if there are out-of-the-box connectors between the CMS and your other systems, and if there aren't, make sure your technical team evaluates the API documentation and feels confident it can build the integrations the business needs.

What Are APIs?

APIs (Application Programming Interface) allow the exchange of data between different systems and the implementation of processes across those systems. Think of the API as a bridge between applications. If a developer wants to access data from one application and use it in another application, they call the API. The API accepts the request, approves or denies it, and if the request is approved, the API returns the data. The API also tells developers what they can request, how to ask for it, how to receive it and which formats it will accept as input or output.

REST APIs tend to be a great option when integrating with third-party systems. REST APIs are a type of API that provides Web Services to access the system remotely. These APIs use internet standards such as HTTP for the communication, and XML or JSON (which, to put it simply, enable text and documents to be human-readable and machine-readable) for the information. Why does this matter? Well, developers can call REST APIs from any application with almost any programming language. They are also extremely simple, lightweight and fast to implement.

[Dig deeper into APIs on the eZ Blog >>](#)

3 Granular User Permissions

Often times staff from different departments in an organization use the CMS, and sometimes external users also contribute content. In order to create a clear separation of duties and ensure the content is secure, it's key to prioritize user permissions when evaluating CMS.

In a CMS with a [role-based architecture](#), an administrator can define highly precise permissions, all the way down to a specific content type or action such as viewing, reading, downloading, creating, editing and deleting content. Perhaps you want to restrict certain users from publishing or deleting content, or maybe you want editors to have full rights over specific sections, or even just specific content within a section. Be sure the CMS gives you this level of control.

Also ask if you can create your own permissions configurations to meet your specific needs, and while this may seem obvious, make sure no one can get access to the admin interface unless the administrator explicitly defines and grants it first.

Workflow Management

The creation, review and publication of content often involves many stakeholders. When evaluating a CMS you should ask many questions about workflow management. What kind of workflow management does the CMS come with out of the box and can you create custom workflows to suit your needs? If you use external translation agencies, are there workflows suitable for their contributions and subsequent translation reviews and if not, is the platform open so your development team or agency can build them? Certain content types may need to be reviewed by Legal prior to publication. Does the CMS provide different workflow paths to accommodate different needs?

Traditional vs. Headless CMS

A traditional CMS covers authoring, storage of content and a web delivery tier, usually relying on a template system to deliver the content on web pages.



A headless CMS, on the other hand, only provides an interface for authoring, a repository for content storage and APIs for content delivery—there is no template system to control the presentation of content on web pages. With a headless CMS, you essentially remove the website itself, including the page templates and themes. There are also

decoupled CMS, which can be used for both traditional and headless approaches.

The right CMS for you depends on what you want to achieve. Is your business only building a simple, 6-page brand site? If so, a traditional CMS will probably be best. Are you interested in building out your online presence across channels such as mobile and desktop apps or connected devices (IoT)? Do you need more flexibility to create custom layouts beyond the templates in your current CMS? If so, a headless or decoupled CMS is the way to go.

The way we see it, even if you don't have complex needs today, you should invest in a CMS that will support your digital experience strategy for years to come. Our take? Invest in a decoupled CMS and get the best of both worlds. Your editors, developers and UX designers will all be happy with your choice.

[Download our eBook on Content as a Service >>](#)

4 Multilingual Content Management

Globalism has changed everything in business. Many brands are international and many more will be in the years to come. Creating specific content for each region can be complex, time consuming and dilute the vision and message of your brand. If you're already catering to an international audience, or you have plans to cross borders in the future, you need to pay special attention to your multilingual CMS requirements. We've put together a list of the most important things to look for in a [multilingual CMS](#).



Out-of-the-box functionality

Many CMS offer multilingual functionality as plugins or extensions. While this can work, it can introduce vulnerabilities. Plugins can be coded poorly and go unmaintained since they are not created by the actual CMS vendor. Malware can sneak in from third-parties, and updates are often few and far between. An out-of-the-box multilingual feature will have less security vulnerabilities and will likely be better maintained because it is a product of the vendor.

Translatable content

By creating a standard content structure in the repository that applies to all of your sites, your editors can quickly and easily build and maintain various digital properties. Prioritize a repository that enables you to store translations of content centrally in the same location in the admin interface as the

root, or parent, content. In other words, if you translate product page A into two other languages, the translated content should be stored in the same location in the repository as the original product page, as a translation.

This makes managing your content across languages and sites much easier, streamlines training editors who manage content on multiple sites and facilitates smoother collaboration among editors. It also helps simplify the developer experience. If a feature works in one language, it will most likely work in other languages (but of course, be sure to test). Be sure to ask which languages the CMS supports. Many CMS do not support exotic character sets such as Mandarin or Japanese, and even if your organization isn't doing business in these markets today, you want to invest in technology that will support you for years to come.

Multilingual versioning

When editing, regardless of the language, it helps to go back and see your previous edits at times. This is no different in a multilingual environment, perhaps it is even more important. Make sure your versioning system supports all the languages you plan on using in your multisite setup.

Language-specific permissions

When you have a variety of editors working with the same CMS, it's best to make sure they can only edit content that pertains to them. Pick a CMS that has the ability to apply user permissions based on language. This will help make your editors' jobs easier and ensure content is secure.

Language support for the admin UI and site structure

Your CMS system should make it possible to localize editorial user interface because you may have editors from different countries speaking different languages. You should also be able to translate elements that make up your site structure such as error messages, which may be managed outside of the repository.

Don't Forget the Developer Experience!

When it comes to CMS, good developer experience means first and foremost that the software is easy to download and install, and that developers can customize and extend the system in a simple and efficient way, following the best practices and modern trends of software development. A CMS truly built for developers will have open and well designed APIs, thorough, up-to-date documentation, tools, an active and agile release schedule and support for continuous development. The developer experience also covers how developers upgrade the system to new versions. Backward compatibility and a clear upgrade path are key.

Look for a CMS that is based on a trusted and widely used framework—this will make it easier for developers to learn and build on top of the platform, and it'll open you up to a wider pool of developers and agencies that can help you implement your projects, whether you find partners or hire in-house. Also be sure to get a feel for the community—if the CMS is open source—and see what kind of plugins, modules, extensions or bundles are available to the public.

Evaluating CMS Frameworks

Whether you are looking for a platform based on PHP, .NET or Java, for example, you should look beyond the programming language and consider which framework the CMS uses.

CMS vendors and communities use frameworks to help speed development and extend functionality of what's out of the box. Some CMS are based on their own frameworks while others use existing frameworks in the market.

Make sure the CMS, and thus the framework you select, is easy to build upon and that there's an ecosystem of developers behind it. Look for a well-known and recognized framework that is sustainable and constantly evolving with new ideas, plug-ins and users. Don't forget to ask about available support.

For a closer look at our views on the technical side of choosing framework, read our blog post ["Why PHP and Symfony and not Java?"](#)



5 Multisite Support

Departmental leaders commonly share the misconception that launching a website built on a CMS dedicated to their site and their site only, and managing the project in a very independent way, will save their organization time and money, while affording them the freedom they need. In reality, the result is exactly the opposite.

Managing multiple sites with multiple CMS increases the need for training, limits collaboration, dilutes the brand identity and leaves room for error due to burden of testing multiple systems. It also makes testing and site updates more complex and puts more burden on the business, hurting the efficiency of your editors and developers.

If you're looking to alleviate all this pain, a multisite CMS is an important consideration. With a multisite CMS, it's easier to reuse content, share workflows, consolidate training and technical development resources and implement organization-wide templates and style guides. Your developers can streamline testing, simplify integrations and build sites faster while your marketers and editors can increase brand consistency, foster collaboration and share content across sites. Administrators also

stand to benefit, as they can manage users and permissions across sites from a single platform.

Do your organization a favor and start discussing CMS consolidation across divisions. We know it isn't simple but it'll pay off immensely down the road.



A multisite setup is much more scalable than managing individual websites with multiple CMS. Development is simpler. Testing is easier. And implementation is faster.

Editors Deserve a Better User Experience, Too!

For marketers and editors, a great user experience means first and foremost that it's easy to navigate the CMS' user interface and publish content fast. The interface should be logically organized, easy to understand and pleasant on the eyes. Marketers also need plenty of functionality to make their daily work more efficient. Some key requirements usually include the ability to navigate the repository easily, preview content across devices, create dynamic landing pages and schedule their content in advance.

Don't forget the other users within your company. Some, like HR, Product Management, Legal and your external guest contributors, may only use the CMS occasionally. If they use it once a month, will they remember how to login and post content? Having to constantly retrain users on these steps is costly and inefficient so be sure to prioritize the user interface design.

"I think the biggest mistake companies make when selecting a CMS is focusing solely on a set of features... What they rarely consider is the actual user experience of the system. I'm talking about how the CMS will be for the internal users, people inside the organization such as marketing and human resources."

- Paul Boag, User Experience Expert, Author and Founding Partner at Boagworks

6 A Powerful Web Template System

There's nothing worse than having a beautifully designed website render jumbled text and images on a smartphone or tablet. All your hard work goes down the drain if your customers can't see it, or don't see it the way you intended.

Your CMS should provide a template system for producing and deploying web pages on your site. Be sure the CMS makes it easy to implement templates that adapt well to different screen sizes.

The template system should have a templating engine which separates presentation and business logic into two independent parts. With this approach, your front-end developers and designers will have the flexibility to create what they have in mind. If these two components aren't separated, the front-end design will be limited, and some designs will be impossible to implement.

Also check that you have full freedom to control the presentation without having to dive into the code. You should be able to adjust the markup—the tags that describe the document's logical structure and indicate how the file should look when it is displayed—without touching the HTML or CSS. This typically comes in the form of a WYSIWYG editor, though we are seeing demand for contextual online editors that provide users with different sets of tools geared towards the editor's task at hand. Ask how each vendor handles this requirement, and check if the online editor is customizable.

You should also evaluate the templates provided by your CMS by default. There may be some helpful responsive templates out of the box, which you can use, customize and adapt.

Responsive Design vs. Adaptive Design

These days providing users a smooth mobile experience is critical for business of all types. Here are two approaches to consider, and remember, you can also use a hybrid of the two on a single site.



With responsive design, you have a website that automatically fits the size of the screen no matter the device. The site adjusts based on client side criteria such as display type and screen width and height. Usually, there's less work involved but on the other end, you give up some control and cannot customize the user's experience precisely for a given device. Page load times can also suffer since all elements on the web page are loaded, even if they aren't visible to the user.



An adaptive design approach is a bit like retrofitting your website for specific devices. You have more flexibility in controlling and customizing the look and feel of your site for a particular device. You can serve completely different content based on what kind of device is detected, and serve different layouts. Adaptive design adds complexity and cost since more work is needed upfront and ongoingly to maintain multiple layouts for different devices.

[Learn more on the eZ Blog >>](#)

7 Web Content Personalization

When evaluating a CMS, you should also consider whether you want to implement a web personalization strategy now or in the future. Some CMS provide personalization capabilities out of the box while others will integrate a third-party cloud service—some CMS even offer connectors for major SaaS solutions.

In our view, the most important thing to decide is your strategy and operational readiness, then start digging into the technology. We go into great detail on this subject in a separate eBook, *A Primer on Personalization*.

Now, you might be asking why you should be considering web personalization. A personalization strategy implemented well can deliver immense benefits for your business and customers alike. An effective personalization strategy can increase traffic to your site, keep your users engaged and ultimately increase your revenue. As for your customers, a personalized experience helps them find what they are looking for faster, reduces visual clutter and creates a bond with your brand.

Personalization is most commonly championed in the retail/B2C space. If you are a retailer, you can recommend similar products based on what the user has purchased or evaluated, to drive revenue. Publishers can also benefit from implementing a web personalization strategy. By delivering

relevant content—articles, videos, image galleries, etc.—they can increase content consumption, convert freemium users to paid content and drive engagement of new content bundles.

Today we are also seeing much more subtle personalization based on machine-learning of user habits or comparisons to other, similar user behaviour. The structured content model we discussed at the beginning of this eBook is key to this delivery, allowing you to try many different approaches to personalization such as changing text, headlines or images for a specific context or persona.

Be sure to make personalization transparent to the user so that useful and relevant content is delivered without feeling like an invasion of privacy. Users would much rather feel like they are discovering personally relevant content than being “sold” to.

Personalization is usually implemented through an automated and rules-based approach to database segmentation. Personalization is based on user behavior (clicks, page views and purchases, for example) or demographic data (location, job role and industry, for instance self-identification), and in many cases a personalization strategy will encompass both types of data.

[Download our eBook *A Primer on Personalization* >>](#)

The screenshot shows a section titled "Customers Who Bought This Item Also Bought". It displays five book covers with their titles, authors, and brief descriptions. Each item includes a star rating and a "Prime" badge.

| Book Title | Author | Description | Rating | Price |
|-----------------------|------------------|---------------------------------------------------------------|---------------------------|----------------|
| Sweetbitter | Stephanie Danler | A novel | 4.5 stars (463 reviews) | \$15.45 ✓Prime |
| Before the Fall | Noah Hawley | | 4.5 stars (1,758 reviews) | \$13.00 ✓Prime |
| Modern Lovers | Emma Straub | New York Times bestselling author of <i>The Lost Daughter</i> | 4.5 stars (188 reviews) | \$15.60 ✓Prime |
| Truly Madly Guilty | Liane Moriarty | An #1 New York Times and #1 International Bestseller | 4.5 stars (662 reviews) | \$16.19 ✓Prime |
| The Woman in Cabin 10 | Ruth Ware | Author of <i>Die a Little</i> , <i>Don't Be Tardy</i> | 4.5 stars (232 reviews) | \$11.44 ✓Prime |
| Girls on Fire | Robin Wasserman | | 4.5 stars (97 reviews) | \$17.86 ✓Prime |

Page 1 of 17

Amazon recommends products based on the user's behavior as well as the behavior of other users who bought the same items. This concept is based on establishing patterns, theorizing that users who buy the same product are more likely to have mutual interest in other items.

Lessons Learned

We hope you enjoyed this eBook. If we did our jobs right, you feel more confident in your CMS requirements than before and you are better positioned to make an informed decision on your next platform. Before we go, we'd like to provide a summary of the key takeaways.

- 1.** Look for a CMS with a flexible content repository and a strong separation between content and presentation that allows you to structure your content so you can reuse it and repurpose it for any channel.
- 2.** Ensure your CMS has open APIs so you can deliver content across channels, platforms and devices and integrate your CMS into the rest of your technology infrastructure to create new business efficiencies.
- 3.** Be sure your CMS provides a template system and full freedom with the markup itself without having to dive into the code.
- 4.** Look for a CMS that provides multilingual content management in a centralized repository.
- 5.** Put a high priority on a role-based architecture for user and permission management, allowing you to define permissions at a granular level.
- 6.** Consider implementing a web personalization strategy, and ensure your CMS can deliver relevant content to the right user in the right context—or that it can smoothly integrate personalization solutions—to drive user engagement and generate revenue
- 7.** Don't forget about the developer experience and the editorial user experience—the right CMS for your organization isn't just a matter of features.



About the Authors



Henry Kogan is Content Marketer at eZ Systems. He has over six years experience writing for technology and finance. Henry began his career as an Associate at WeiserMazars LLP's Financial Services Group.



David Kross is Director of Marketing at eZ Systems. He leads strategy and execution of eZ's product, content and digital marketing programs and edits the eZ Blog. David has 12 years of B2B marketing experience with a focus on enterprise software and SaaS solutions.

Designed By



Hakan Ertan is eZ Systems' Graphic Designer. He is the creative mind behind the company's visual identity. For the past decade, he has lead award-winning projects for Saatchi & Saatchi, TBWA and BBDO. He specializes in advertising art direction, illustration, conceptual design and interactive assignments.

A special thanks to Richard Jones, Technical Director Content Practice at Inviqa and Roland Benedetti, Chief Product and Marketing Officer at eZ for their contributions to this eBook.

eZ in Brief

eZ Systems is a global content management platform provider that is dedicated to helping businesses maximize the value of their content. Our software simplifies the way enterprises create, deliver and optimize their content in order to engage customers. Thousands of organizations rely on eZ to deliver digital experiences that foster business growth.

eZ Facts

- eZ Systems AS was founded in 1999 in Norway
 - 15+ years experience as a commercial open source provider
 - Supports 500+ enterprise customers in 25+ countries
 - Ecosystem contains 80+ business partners and a community of 45,000+ members
 - Over 5 million downloads and 250,000 installations
-

eZ Links

- » Website: www.ez.no
- » Blog: www.ez.no/blog
- » Case Studies: www.ez.no/Resources/Case-Studies

About Inviqa

In a world where digital is redefining business, choosing the right technology partner is fundamental to achieving your goals.

Inviqa is the technology partner you can trust to put your business first. A specialist in content management systems and ecommerce, Inviqa helps pioneering brands and organisations to engage, convert and retain their customers through digital innovation.

Inviqa has offices across the UK and Europe and employs more than 200 professionals, including software engineers, business analysts, project managers, and designers - a combination that has supported some of the largest organisations worldwide.

Visit inviqa.com to learn more about our work and how we can support you.



| www.ez.no

| info@ez.no

| [@ezsystems](https://twitter.com/ezsystems)