The Web CMS is Dead

How an agile content platform can deliver every site and channel

INGENIUX

Content beyond boundaries

There are many content management systems designed for managing websites, but the future of content management goes beyond web experience. The modern CMS needs to power the total customer experience, from marketing to customer success, from partner enablement to employee collaboration. Content permeates every business touchpoint. Traditional Web CMS applications struggle to meet the requirements to deliver content on multiple channels and experiences.

Web CMS is broken. We've lost our way by building complex CMS-powered websites that over promise and underdeliver. We've valued acronyms over people – making platforms that are too hard to update and implement.

You need another, more agile option.

You need a CMS that puts content first by providing the structure and intelligence you need to reuse and target your content easily. You need a content delivery system with the flexibility to deliver content anywhere, in any format – from dynamic to headless and beyond.

On the flip side, the API only headless CMS' don't provide you the tools you need to deliver a complete web solution.

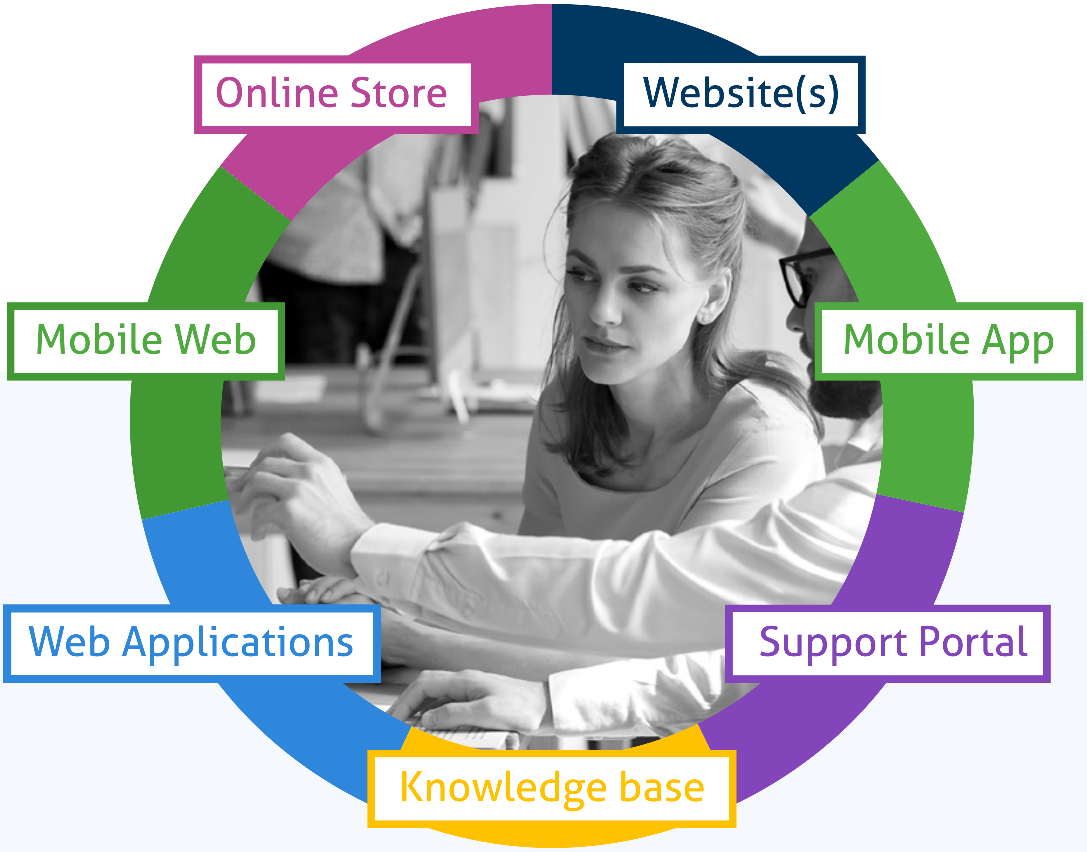
This guide will help you navigate these changes and learn how you can decouple your CMS from your website to liberate web development and expand the reach of your content.

What’s Behind the Shift Away from Traditional Web CMS

We’ve been headed this way for a while. Web content management systems evolved from serving static HTML files to the website to serving dynamically generated website content to the website and CMS-based publishing channels.

With backend content management administration tightly coupled with the front-end display in the traditional CMS, all your publishing channels live within your CMS. Additional modules provide analytics, search, social media, personalization, even email marketing – a complete web experience platform in a box.

But here’s the thing. You are no longer dealing with only your website as a channel to engage customers and prospects. The website is still essential, but it’s one channel of many. Mobile applications, customer portals, multiple websites, secure applications, voice-enabled search, and more are channels separate from your CMS, but requiring the content that you manage in your CMS.



From Page-Based to Content-First

Tightly coupled Web CMS platforms assume you will build all your experiences within the platform, which is too complicated and often impossible. They assume the focus is primarily on your website and creating web pages within that website.

But as you add new channels to your digital experience strategy, you find yourself shifting from a page-based content strategy to a content-first strategy. A content-first approach enables you to identify your content needs for multiple channels and focuses on how you can reuse content across those channels.

Content Curation and Orchestration

There are many departments and groups across the organization that are creating content you can use in your digital experiences. Forcing everyone to use the same content management system isn't going to work. Instead, you want to curate content from across the organization, making it available through your CMS.

Your CMS will need to provide orchestration capabilities to manage the content, including connecting content systems, implementing a shared taxonomy and metadata strategy, ensuring security and permissions are correctly applied, and more.

Content Strategy

While it may not make sense to use a single content management system for the entire organization, it does make sense that a centralized content strategy support the needs of every digital experience to ensure a consistent, relevant content for the brand.

Personalization/Dynamic Experiences

Creating the website experience within the Web CMS was the typical approach for many organizations, which is why many CMS incorporated additional modules to support a more dynamic, personalized experience. But there are many new technologies available today that are opening the door to hyper-personalized experiences, including front-end development technologies like Angular and React. These technologies exist outside the CMS and need a way to access the content stored within it.

The Traditional Web CMS Has Met Its End

If you follow the research of well-known analyst firms Forrester and Gartner, then you know that we have entered a new era of content management. Gartner has ended its Magic Quadrant report for Web content management systems, indicating that web content management has become a commodity, with very little differentiation between products. Gartner points to the desire for organizations to support the consumption of content outside the website.

“It’s the birth of WCM to a new definition of a content management system. Because now it’s not only the Web, obviously, that we need to tackle when you have Alexa, chatbots and different devices, modalities and channels. Content is not dying. It’s just turning into an original idea of a content management system that’s intended for different content types and now for different channels.” - Irina Guseva, Gartner lead analyst and lead author of the former WCM Magic Quadrant and senior research director focusing on WCM and DXP (source CMSWire).

Forrester has referred to it as “the end of Web CMS as we know it,” pointing to the need for the CMS to support more channels and experiences apart from the website.

“Firms need to pivot to a decoupled architecture — our DX stack — to support a broader digital experience mandate enabled by key content, data, and transactional building blocks.” - Forrester, Agile Content Curation And Orchestration Will Redefine CMS.

> You need a content management system that is flexible enough to meet the technology and multi-channel demands of today’s digital experiences. You need an agile content management system.

The Need for Agile Content Management

The evolution of web content management technology has happened in tandem with the evolution of digital experiences. Many WCM vendors have been slow to make the changes required to support continually evolving customer demands, and the truth is, it's tough to do if you are still thinking in terms of traditional web content management.

To support the needs of companies and their customers today, and in the future, an agile content management system is required. A CMS that enables your company to keep up with changes without having to re-platform, rebuild, or completely redesign digital experiences every time something new comes along.

An agile content management system differs from traditional CMS in several important ways.

It's not an all-in-one CMS.

With a traditional CMS, everything you need is found within the CMS, whether you use it or not. And often, some of those capabilities provide only basic features. An agile CMS focuses on the capabilities required for creating and managing content that you can use across all your customer channels and devices.

An agile CMS provides an intelligent content model that supports the structure and organization of content in a way that describes it – ensuring that all channels that use it can pull what they need and display it any way they want. This semantic definition enables the CMS to adapt the content for multiple outputs and formats and makes it easier to find by both machines and humans.

It's decoupled.

A traditional CMS is tightly coupled – you can't separate the management of content from its display. An agile CMS decouples content management from content delivery. You can still create delivery tiers within the CMS, like your website, but those delivery tiers are separate from the management tier. To support the separation of management from publishing, an Agile CMS includes a content delivery framework that can render dynamic content and provide other services.

It's headless.

An agile CMS also includes a headless API that supports the delivery of content to separate websites, applications, and other publishing channels. You manage your content in the CMS, but that's all, and you access that content using a REST API.

> Simply put: An Agile CMS is Intelligent Content + Decoupled CMS + Hybrid delivery.

Agile CMS vs. Headless CMS

It's easy to see the difference between traditional content management and agile content management. But what about the difference between Agile and Headless? Some might think the difference is more subtle when it's evident that agile provides many more benefits than a purely headless model.

A Headless CMS separates content managed in the CMS from the site or application displaying that content. A Headless CMS does not provide a front-end delivery tier. Instead, it provides an API that developers use to pull content into their custom-designed websites and applications.

Headless has many advantages: fast to set up and integrate, greater control over the front-end design and development, true dev-ops support, and often better information security practices. But most headless CMS' are simply a content database. They have no understanding of where the content will be used, how it will look, or how to govern it. That's why it's only part of an optimal content management solution.

Agile content management provides everything you get with a headless CMS and more:

Delivery flexibility.

With a hybrid CMS you get to choose where and how to deliver your content. You can create your website using the content delivery framework within the CMS, enabling your marketing team to easily manage the web experience by creating and publishing new content and landing pages, changing existing web page templates, and more.

But you can also create separate digital experiences that use the content managed in the CMS. For example, you can store help documentation for your credit card application and pull that help content into the app using the headless API. Or you can develop a mobile app that gets regular content updates from the CMS for news, announcements, and other content.

The choice is yours. Mix and match your delivery channels with all having access to the same content ensuring a consistent experience across all your channels.

Robust governance tools.

An agile CMS provides not only a content database but also the tools to ensure that content is correct and aligns with your brand. It includes a set of governance tools such as fully integrated content workflow, grammar and spellcheck, and a broken link checker.

Faster time to launch.

We mentioned that one of the benefits of a headless CMS is fast set up, but an agile CMS provides faster time to market. With a headless solution, you have to build your front-end separately, dealing with development efforts that can be fast or take a lot of time, depending on what you are building. With an agile solution, you can leverage templating features and the delivery framework within the CMS to create and deploy web experiences quickly.

Built-in preview.

With a headless CMS, you have no preview for how your content will look when it reaches its destination. Content authors enter and update content blindly, going back and forth to the published version to ensure the changes are correct. This disconnect can create inefficiencies in content creation, quality assurance, and dev-ops processes because authors need to render the content in the project, see it, make changes, and redeploy to meet branding, layout, and formatting standards.

With an agile CMS, you get contextual editing with a high-fidelity preview of how content will render on a page, screen, or device. The authoring experience is available using In-Context Editing for headless content. In-Context Editing, or ICE for short, allows authors to edit content using a set of editing controls available in the preview of the Web CMS, eliminating review loops.

Agile CMS Use Cases

Let’s look at some use cases for agile web content management.

Modern Sites

Modern websites focus on delivering dynamic, personalized experiences. Many brands are providing these experiences using front-end frameworks like Angular, React, and Vu. These lightweight JavaScript development frameworks enable developers to create highly customized websites. But they are only for front-end development. To serve up dynamic content, they connect to a content management system, usually through a REST-based content API, and pull in the content they need, formatting it on the front-end.

Of course, not every company is building their website this way. Some want a website that is easy to update, including adding new templates for web pages or new sections. With an agile CMS, you have the option of building your website on the CMS platform directly. In this case, the delivery of the website is still decoupled from the management of the content, but the CMS provides the delivery framework, making it easier for marketers to update their websites as necessary.

Portals

Customer support portals, knowledge portals, student portals, and other types of portals provide users with a combination of content, functionality, and links to external applications. An agile CMS can provide the ability to curate content and data from many different sources, feeding it through a central repository into the portal environment.

Mobile Apps & Business Applications

How many mobile apps are on your phone? Many of them offer up content that you can manage in a CMS, making it easy to update the content without having to update the mobile app itself. Web-based applications are another example of an application that can connect to your CMS to pull in content that is updated regularly.

Both examples leverage the headless API available in an agile CMS, pulling content into the application or mobile app through a secure content API.

Technical Content and Documentation

Technical content used to be the domain of technical communicators only. But much of this information is made available today to customers who want to self-service their questions and issues. Customer service and support portals provide this self-service capability, and these portals pull in technical content in the form of XML and DITA to supply searchable knowledgebase content to customers directly.

6 Tips to Get Started with an Agile CMS

1. **Create a Content Inventory:** The first thing you should do is an inventory of all the content in your organization. Document what the content is, what system it is stored in, who owns and manages it, who uses it, where it's currently used, how often it's updated, and so on.

   If you can't do it all at once, start with a single department and build your inventory slowly as you work on projects that require additional information found in other locations across the organization. Look for overlaps in content and note if that content is identical or has important differences. You'll want to see if you can merge related content and delete ROT (redundant, obsolete, and trivial) content before you go further.

2. **Develop a unified content strategy.** You need to think outside the page-based model of traditional content management and design a content strategy that focuses on the content itself, outside of where it's published. Map out the content in your inventory and start defining the high-level taxonomy, including topics and content types. Define metadata attributes to apply to content topics and identify any relationships between each content type.

3. **Create a Unified Repository:** Aggregate content created across the organization, including marketing, sales, support, engineering, and customer service, and add it to the repository for centralized access following the metadata strategy you defined. Remember, you don't have to create all your content in a single CMS, but making it available through a central repository means everyone is working from the same content.

4. **Identify publishing channels:** Map out all the places where your content is published currently, what content they publish, who owns the channel, how often it's updated, and so on. Add in projects in progress that will use content from the CMS as well, making sure any updates to the content inventory and taxonomy are completed to keep everything up to date. For each channel, note how the content is published to that channel, enabling you to identify the types of deployment models you need to employ, including headless API, push-based delivery, dynamic delivery using the CMS server technology, and other approaches. You will add new channels as your digital experience evolves, so make sure you are always updating the channel information as well.
5. **Integrate third-party tools as required.** The digital experiences you offer will require information from other key systems in the organization, including marketing automation, commerce, apps and data, ticketing, knowledge, CRM, and more. The value of your digital experience increases with each application that you add because it ensures you can provide a consistent content experience for your customers.
6. **Communicate, Educate.** It's critical to communicate your new agile approach to content management and content delivery. Define the guidelines and best practices for creating and managing content from now on. Search-enable your new content repository so users can look for existing content before they create it from scratch. Outline the processes to set up new publishing channels and outline the framework to connect to the repository to access the content they need. Continually update and promote newly available content, new channels created, and new experiences defined.

Are You Ready to Go Agile?

It's amazing to think of all the digital experiences you can create today to engage and inform your customers and audiences. Trying to manage all the content for those experiences in a traditional Web content management system is impossible. That's why the Web CMS as we know it is dead.

It's time for an agile approach to managing and delivering content for all your sites and channels. It's time for an agile CMS.

ABOUT INGENIUX

Ingeniux is the leading provider of web content management and digital experience software. We enable organizations to orchestrate the entire customer experience from acquisition through to sales to support and service, across any device, application, or website.

We build content management software with an unparalleled focus on the content itself. The Ingeniux CMS is designed to manage and deliver modern websites, customer support portals, online communities, and other customer touchpoints.

We believe in intelligent "structured" content. We design our software to enable content reuse, enable true mobile and multi-channel content delivery, and insightful content discovery. Our unique content-as-a-service capabilities deliver content into web and mobile applications, and other key channels.

Ingeniux software is available as a fully managed software service or an on-premise application. Ingeniux delivers unparalleled service and support to customers worldwide.

To learn more, visit us at www.ingeniux.com.

INGENIUX

PO Box 21466
Seattle, WA 98111

info@ingeniux.com
877 445 8228