

The modern website strategy guide: how content infrastructure is accelerating digital products



Contents

Introduction	3
Modern websites = digital products From web <i>projects</i> to web <i>products</i> Software best practices also apply to websites	4
Content infrastructure: the heart of the modern website Limitations of traditional CMS Content infrastructure: designed for the modern tech stack	8

Conclusion

Introduction

The way people experience the web has radically shifted. Today's websites are no longer simply a collection of published pages—they are full applications in their own right. The best websites are engaging, personalized digital experiences that are driven by data and enhanced by ever-changing new technologies. Accessed by users on hundreds of devices and platforms, websites now demand as much attention and iteration as any core product.

In response, business leaders are thinking about their websites in a whole new way. They recognize that digital experiences must not only delight users, but also maintain a close alignment with business strategy and external variables such as advances in technology, new customer needs and time-sensitive events. Websites must consistently deliver value to customers, partners, employees and other stakeholders. As more enterprise organizations make software development a strategic priority, they are taking a product-centric approach to building and running their web properties alongside other software initiatives.



At the end of each section, we provide links to relevant content, both business-focused and technical, to help you research concepts further.



Modern websites = digital products

Traditionally, engineering organizations built or rebuilt their websites as one massive project with a set budget, timeline and launch. Post-launch, they might update website content incrementally, but usually little investment is made in the site as a whole until the next major "site refresh."

For the company, this means another costly and resource-intensive project that usually happens every several years. In the meantime, the website's users, as well as the market and business, all move forward, leaving the website to stagnate.



From web project to web product

The product-centric approach to web development is different. When websites are treated as products, they get the ongoing focus and resources needed to keep them relevant and valuable to both users and the business. There is no set end date as with other engineering projects.

The website product is a living, evolving software application with its own strategy, road map, and team. This product model enables the organization to regularly release new site features or updates so the website can continue to deliver value as the business evolves. In this scenario, the technology and resource investment may be ongoing, but the end result delivers far better returns as the website works harder for the company.

Built by cross-functional product teams

So, who builds website products? In the old model, the website project team included various specialists with a short-term focus on specific aspects of the site, such as design, marketing, development, or testing. However, they typically worked from the perspective (and directive) of their siloed organizations.

Product teams, on the other hand, are inherently cross-functional. They include the same specialists, but everyone is dedicated to their product's mission. Working together full-time, product teams have an ongoing focus on the lifecycle of the website, its users, and its value to the company.

Product teams enable a more cohesive product evolution and a stronger focus on customer and business needs.

Product managers are website product leaders

Website product teams are led by product managers, rather than IT project managers as in the engineering project model. According to McKinsey¹, today's product managers are the "mini-CEO" of their product and play a pivotal role in its ultimate success. They are typically part technologist, part business manager, and are responsible for the product's viability in the marketplace.

Product managers guide the team's product vision, strategy, and roadmap, and they represent the product to stakeholders and others across the organization. With a relentless focus on the user, the product manager ensures that the website delivers the best possible digital experiences across the customer journey.

Read more:

Content Management & Strategy Survey 2018



Software best practices also apply to websites

As organizations reinvent their websites as products, they are building their sites in the same way as they build other software applications. Websites can also benefit from the speed, flexibility, and quality associated with using modern architectures and development best practices.

Agile development

To stay competitive, most companies are racing to address dynamic forces in their industry landscape. According to Forrester², "as software takes a central role in modern business models, application delivery capability has become the essential enabler of an organization's digital business strategy." Speed and agility are essential-product teams must be able to respond quickly and update or iterate their product frequently. Software delivery methodologies that enable this hyper-responsiveness, such as agile development, allow teams to ship website updates continuously. Fast, fluid website iteration, with numerous daily pushes to production, also keeps users engaged in ever-changing, relevant content and features, and supports evolving business objectives.

"Long-term strategic planning and lengthy product cycles have given way to a feedback-driven cycle in which customer needs and competitive threats change quickly."

"Faster Software Delivery Will Accelerate Digital Transformation," Forrester², April 2018

Microservices

To further increase product agility, product teams are moving away from building websites as monolithic systems that mushroom over time. A monolithic code base eventually becomes hard to work with due to its size and complexity– an issue often exacerbated by bloated code. Updates become tricky as changes to one part may unintentionally break another, resulting in longer development and test cycles.

Instead, modern software architects favor a microservices approach that composes a web app using a constellation of separate services– essentially mini-applications–all running in harmony to create a unified user experience.

For example, a website could be composed of a messaging service powering chat, an authentication service for account login, and a data handling service, all seamlessly tied to the front end. In this model, changes can be made to a single service without impacting the others for faster iteration or troubleshooting.

"The adoption of Lean and Agile practices is a journey, not a destination. While they offer great principles and practices, they are substantially transformationenabling frameworks."

"Faster Software Delivery Will Accelerate Digital Transformation," Forrester², April 2018

Serverless architecture

Another important trend in software development is serverless architecture, which helps product teams achieve scale faster. A serverless architecture does not mean that there are no servers running code, but simply that development teams don't have to deal with provisioning and maintaining servers.

This model provides scale on demand, even when a website hits production peaks, and typically associates operating costs with how much code is actually invoked. As product teams don't have to worry about server administration, their engineers can focus on writing code and developing the best product experience.

Read more:

GraphQL and Serverless: Where Cloud Computing is Heading

Serverless Architecture: Going Serverless with Contentful

From legacy websites		To digital products
Collections of templated pages, managed by a monolithic CMS backend	>	Content powered by microservices, employing multiple architectures, and adapting to multiple touchpoints
Built with a waterfall approach: First design, next code, then populate content, finally publish	>	Built with an agile approach: Any part can be worked on and quickly changed independent of the others
Friction and delays at handover points between team members	>	Everyone works together in parallel
Massive homegrown systems try to do everything at once, such as manage assets, author content, run analytics and handle payments	>	An API-driven stack integrates individual best-in-class services managed by third-party vendors who are experts in their field
Implementation teams tasked with customizing an off-the-shelf product	>	Cross-functional, dedicated teams work together to build a content model and translate their frontend layer into an efficient backend

Content infrastructure: the heart of the modern website

Like any digital application, content drives the website experience. Whether you're building a global corporate site, a short-term campaign microsite, an ecommerce business, or a customer portal, how you publish, update and manage your content is critical to website success.

Consider these sometimes competing agendas:



Your end users expect a regular cadence of fresh content; this keeps them engaged and encourages them to return.



Your developers want to focus their talents on building software without the distraction of CMS complexities or hardcoding content.

ſ	
	ŇТ

Your content creators, editors and other stakeholders want self-service control over their narratives, with a simple system to input and push content live in real time.

Limitations of traditional CMS

In the past, companies turned to CMS solutions to power content in their websites. This was appropriate in an era when websites were simpler, however, today's sophisticated web applications—and the product teams that run them—need much more.

Legacy CMSes were designed for publishing posts and pages, not shipping software and apps. These CMSes were born in a page-centric world, with rigid data models and cumbersome presentation layers. In addition, significant IT time and resources were needed to operate a monolithic CMS that was built to run on-premise or self-hosted. On the content side, developer resources are needed to create and update CMS page templates, taking their focus away from building a great user experience. Content editors have to wait to publish new content, and there's little flexibility in managing content experiences across the site. Both developers and content editors struggle with the limitations of CMS, wasting time and impacting the product team's ability to deliver new website experiences quickly and efficiently.

Content infrastructure: designed for the modern tech stack

With the rise of cloud computing, a wave of new technologies has emerged to address the needs of product teams for shipping and managing content. Often called "headless CMS" or "Content Management as a Service," these solutions offer cloud-native, API-first platforms based on structured content models. They are programming language-agnostic and integrate seamlessly into modern application architectures.

At Contentful, we think headless is a good start, but there's more needed for teams to realize the power of platform software. Our "content infrastructure" includes APIs for content management, previewing, images, and display to deliver on complex and unique demands for modern apps and websites. Content infrastructure is purpose-built for the modern tech stack. Like other services within the stack, such as maps, payments or messaging, content infrastructure is a specialized Platform as a Service (PaaS). It powers the flow of content between any number of applications and a central content repository, and provides tools to easily access and manage content within applications. Contentful is one of the industry's leading content infrastructure solutions.

How Contentful empowers developers and editors

Contentful makes content "portable" by allowing your team to create modular, reusable content components that can be repurposed for any website or application. The platform's APIs govern how content can be accessed, viewed, handled, and delivered, as well as a set of content delivery networks (CDNs) for speedy delivery to end users. Moreover, Contentful is a fully managed service, so developers don't need to build and maintain their own content infrastructure.

Free from the limitations of legacy CMS, content infrastructure offers a way of modernizing your content operations and empowering your entire product team. For developers, Contentful provides a central hub for structured content, as well as a suite of powerful management and delivery APIs, webhooks, tools and software development kits (SDKs). These enable developers to quickly build content modules and integrate them into any website.

For content creators and managers, such as marketers, support agents, translators, legal teams and others, Contentful provides a self-service authoring interface that allows them to create and ship content themselves, without engineering support. Content creators can easily create new customer touch points, preview content and push updates live. Contentful optimizes content operations across your team, so both developers and content creators can build, ship and iterate digital products faster. In addition, the scalable platform gives you a future-proof way of building an ever-growing portfolio of new websites and web experiences.

Read more:

How 14 Companies Think Outside the CMS Box

The Need for Speed: Get Digital Products to Market Faster

Blueprint for the Modern Website: An insider's guide to deploying content infrastructure

Conclusion

Contentful was created to meet the changing demands of modern website users. Where companies used to view websites as large, one-time projects that would eventually be handed off to the marketing team until the next big revamp, they're starting to treat them like the dynamic products that they are.

To do this, companies need content infrastructure—a modern tech stack built around APIs and flexible architecture—to replace the traditional, monolithic CMS. Contentful gives stakeholders everything they need to make their website work as well for them as their core product. To learn more about how Contentful can help your business, please <u>contact us</u> to talk with an expert or visit <u>contentful.com</u>.

C contentful

Endnotes

[1] McKinsey, Product Managers for the Digital World, May 2017

[2] Forrester, Faster Software Delivery Will Accelerate Digital Transformation, April 2018