



Sustaining the drive to digital

Six ways legacy platforms can slow you down as you scale



Contents

Introduction	3
Infrastructure	5
Software architecture	6
Content modeling	7
Channels and microsites	8
Usability	9
Configuration and customization	10
Conclusions	12



Introduction

The Covid-19 pandemic has caused a massive acceleration in the quest for digital competency, and companies are realizing that they need to move rapidly into a digital-first model – or risk getting left behind in the dust. In some ways, this makes change management easier. Previous digital transformation projects could only take baby steps to overcome organizational inertia or compensate for the fact employees would have to balance old and new processes simultaneously.

In the past, the rationale for marketing-led projects was the ability to influence growth. On the IT side of the house, cost minimization and risk mitigation was top priority. In the current reality, there is an additional factor to consider: while those metrics are still important, time-to-value has accelerated in priority relative to the metrics centered around long-term ROI. This has become a necessity as many companies rush to create entirely new channels and delivery models where they may not have existed previously.

Some businesses were early to see the potential of digital and either started as a digital-only effort, or pivoted rapidly to take advantage of these larger opportunities. As a result, customer experience has almost become synonymous with digital experience. It's unlikely that a customer interaction does not involve a digital channel in some way; be it product evaluation, ordering or support.

In the digital-first era, the ability to build, iterate and deliver digital experiences — fast — is the new competitive advantage.

Businesses need the utmost flexibility to adapt to changing consumer demands. They also need a solid foundation to start from that propels teams to build, iterate and scale digital experiences faster than ever before.

For example, when the Covid-19 pandemic hit, [Loblaw \(Canada's largest retailer\)](#) was already in the process of consolidating many of its online brand and retail spaces to a single unified content platform. They were able to quickly pivot from marketing content to communicating critical information for shopping in the time of Covid-19, gaining new customers in the process.

Maintaining a predictable revenue stream while capitalizing on a competitive position and organizational agility is key to survival.

The stakes are higher. Even prior to the pandemic, analysts recognized that organizations were seeking far more from their DXP platform choices. Gartner found that only 27% of integrated suite buyers say their martech meets current business needs (compared to almost double this metric for modular DXP buyers). In their 2020 Magic Quadrant for Digital Experience Platforms, they noted, "Organizations that are highly mature in their CX strategies are looking for DXPs that can serve as 'connective tissue' throughout their CX technology stack."

Forrester also sees the evolution of their "Web Content Management" category into what they call "Agile Content Management Systems," which focus on "content hub, collaboration and planning, content-as-a-service/headless, development tools, and channel support" as the key pillars of delivering better digital experiences.

There's a recent trend of legacy vendors trying to sell large platforms as a means of accelerating digital transformation efforts. However, in practice, the value proposition of DXPs no longer matches the needs of buyers. All-in-one DXPs are expensive because they are feature-rich. But as we see from the research, it can be argued whether most organizations actually use those features.

This is a difficult situation to be in during the best of times, but it's far more consequential in this current climate. Similarly, because legacy DXPs are so expensive, organizations actually push even harder the notion they need to be used for all efforts in order to justify their investment. Not only do customers have to deal with software that is less capable and cost-scalable compared to purpose-built tools, but these tools are then forced into larger projects and shoehorned into inappropriate scenarios.

When choosing platforms that scale quickly and sustainably, we've identified six key factors to consider:

- Infrastructure
- Software architecture
- Content modeling
- Channels and microsites
- Usability
- Configuration and customization

Infrastructure



For a platform that needs to scale quickly, infrastructure is a key consideration. Often, legacy CMS systems have particular elements that are time-consuming to deploy, and often do not scale particularly well. In some cases, this would be numerous architectural frameworks and abstraction layers that create slow API and content-serving responses. In others, choices about how content is stored and referenced means that often finding content means going through many layers of hierarchy, or the interface freezing completely if content is moved (and countless references need to be updated as a result). Usually these performance issues are built into immovable decades-old architectural choices and result in vendors adding even greater layers of software (usually caching or asynchronous processing) in an attempt to mitigate the issue.

Of course, this isn't helped by the pre-cloud legacy of most offerings. Some vendors are finally layering caching or content delivery network functionality into their offerings to try to address content performance issues, but this is not a given (and is often an additional cost).

On infrastructure, Gartner goes on to say

“[Organizations] want to invest in cloud-native solutions to deliver digital experiences faster, enable developer agility, application scalability and resilience, reduce technical debt.” Many DXP platforms struggle with these issues.

In contrast, Contentful is designed as a SaaS service from the ground up. This means that concepts like [infrastructure performance, scalability, resilience and security compliance are core to our value proposition](#). Our engineering teams think about obscure and detail-oriented problems like [“How do you scale down your infrastructure without messing with your users' requests?”](#) or how to [resolve DNS caching performance issues with Linux and Kubernetes](#). Massive spikes in traffic due to events like Black Friday or Covid-19 [are handled by our infrastructure without issues](#) and uptime status is [public and subscribable](#).

Software architecture



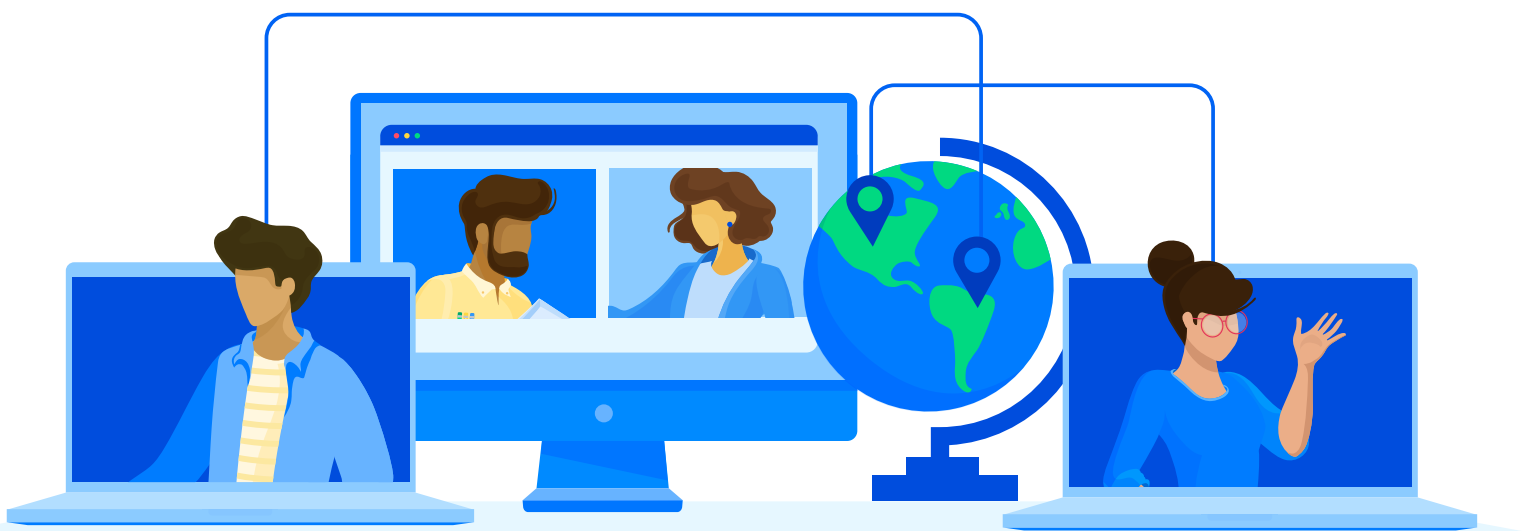
Most legacy DXPs are “coupled” platforms, meaning that generally you are building a single, large (usually compiled) web application which is the sum of your custom front-end code as well as a lot of underlying functionality from your CMS.

In general this is problematic, as users have to constantly keep parity with a codebase that isn't their own (a reason why so many companies are still running old versions of CMS software). It introduces particular challenges when trying to share work across agencies or distributed teams working on the same codebase, thanks to the possibility of conflicting code and deployments.

A legacy CMS platform also requires constant monitoring to ensure that the underlying CMS installation (and module dependencies) are up to date, or else risk becoming vulnerable to intrusion.

This is one of the main reasons Contentful attracts so much attention among enterprise companies – decoupling content from the presentation layer has myriad benefits. Aside from the ability to push content to multiple channels, it also means that teams can scale quickly and gain operational efficiencies without having to worry about underlying versions of other software.

Most importantly, working with underlying content is easier across distributed teams and locations.



Content modeling



Content modeling is the art of creating a blueprint for content and customer experiences – taking into account all visual displays, channels and other “non-visible” elements which help authors find and re-use an extremely valuable business asset. It is one of the core functions of any customer experience implementation, and so Contentful [has created an in-depth guide](#) and comprehensive, [free video training](#) for the task.

In a purely tactical sense, the separation of content from the presentation layer enables things like translation and localization, as well as multi-channel use cases. In a larger strategic sense, it also works well when teams have to think about larger concepts like organizational content strategy and delivering content to the correct channel, locale, persona or step in the buyer journey.

One unspoken aspect of content models is that they tend to change as you add capabilities and channels or learn from the successes and limitations of early models. Despite claiming to deal with larger amounts of content and channels, most legacy DXPs are not able to easily cope with changes or channels.

In the case of Contentful, this is not an issue. It’s extremely easy to create an initial content model –

the Contentful interface is designed to ensure that modeling is a task that business analysts, information architects or writers can easily do without direct developer intervention.

Your model may change over time as you add channels, languages or additional metadata. Additionally – and this is key – despite sharing users and roles in your organization, you can have different models within different environments and spaces in Contentful. These foundational changes can also be easily created, versioned, tested and deployed alongside content with a model known as [“CMS as code.”](#)

Within legacy systems, you are typically limited to one model for your entire implementation. Models built in legacy systems are also typically difficult to manage, since most do not have a mechanism for modifying the model, only adding to it. As a result, legacy CMSes often have many duplicated fields and stacking repeated fields for each new channel or language that gets added, resulting in a frustrating and user-unfriendly copy/paste process.

Channels and microsities



Legacy DXP platforms evolved well before the cloud infrastructure era and accompanying explosion of content channels. As a result, many management and infrastructure choices are tied closely with the actual web and portlet server. Although the benefits of separating content from the presentation layer in a system are well understood, perhaps more important is Contentful's ability to separate repositories from delivery.

The previously close coupling of DXPs and servers has meant that in order to scale, you need to have more and heavier (i.e. expensive) infrastructure at the delivery tier. If you add additional domains to preview granular content or site changes, these often require a server reboot (meaning you need to manage this via failover and other means).

In contrast, Contentful allows you to easily segment out spaces and environments and have an API endpoint for each. Underlying content and roles can be easily shared, but separate teams and use cases can work independently, without having to worry about legacy concepts like server load and configuration-related reboots.

Out-of-the-box integration with modern web development tools means that infrastructure can be deployed quickly and effectively, allowing teams, projects and sensitive content to be developed and previewed independently.



Usability



A platform that claims to be designed to support creators needs to be usable.

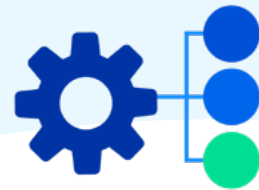
Content creators need to have an application with UI elements that support their tasks (finding existing content, writing, linking assets, editing workflows, etc.).

Many legacy systems depend on “fat clients” in order to drive content creation. This means extremely heavy applications (many megabytes of code), often delivered from on-premise installations that are not cached or co-located near authors (especially true if you have a multi-national team). It’s not unheard of for authors in a legacy system to wait minutes while interface elements refresh. Nevermind the RFP requirement that you can customize every button of the Rich Text editor (which, of course, [Contentful can do](#)) – many authors of legacy platforms would give anything for an interface that just loads quickly!

Contentful is designed to easily support large numbers of authors — and makes it simple to customize the authoring interface on both a deep and granular level. This means that the platform is embraced by content teams and technical teams alike.



Configuration and customization



One of the key rationales for selecting a suite DXP is the multitude of features they offer. In practice, a DXP can usually meet most requirements fairly easily out of the box. However, questions remain: How well do these features actually meet your specific needs? How many require extensive development to do so?

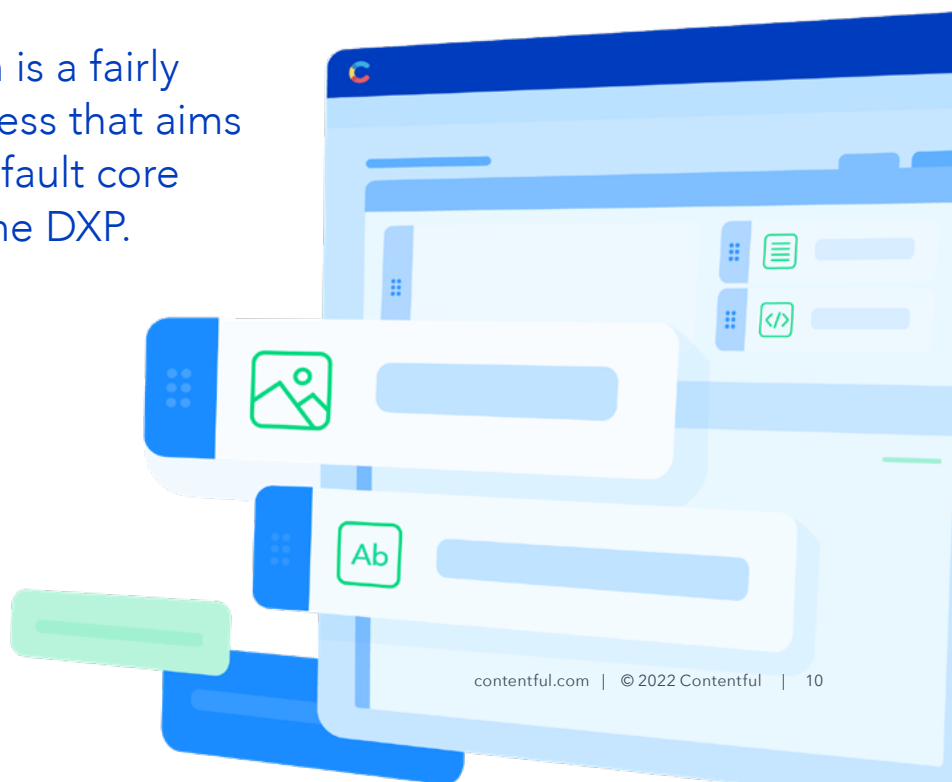
This development effort usually consists of two distinct areas: implementation and customization. Implementation is a less intensive effort that requires less development – mostly centered around crafting front-end visuals or adding custom page components. Customization is a fairly intensive process that aims to override default core behaviors in the DXP (and that’s assuming your DXP even allows that deep a level of system modification).

Contentful provides efficiency gains in both these aspects of development. Its approach to content modeling and the ability to develop in parallel means that core implementation is often faster than a legacy model. However, it’s in customization and development operations where true differences emerge compared to the legacy model.

When customizing a legacy DXP, the development work needs to account for the underlying application, often making upgrades extremely difficult. For example, the majority of implementations are at least one version behind the current supported version, and many are out of support entirely. If you are considering a legacy DXP vendor, ask around and see how many clients are actually running the latest version.

Similarly, areas which are not part of the “core DXP” (such as smaller elements such as form submissions or events) often have a baked-in content model or visual code, which limits possible adjustments to minor and purely visual changes.

Customization is a fairly intensive process that aims to override default core behaviors in the DXP.



Legacy DXPs also run into issues when the implementation grows – but corresponding workarounds can't scale along with it. A number of vendors recommend fairly complex abstraction patterns in order to make it easy to manage the layered problems of different teams working together, and for indicating which bits of code are for specific projects vs. the core implementation.

In contrast, customization in Contentful is easy — and linear. The same content modeling, development principles and patterns apply regardless of the size of your deployment or number of users.

A number of architectural decisions make this possible. As previously mentioned, these include the separation of content and presentation, as well as a microservices-based approach which enables content reuse and platform flexibility.

For customization itself, there are a number of key differentiators:

1. The APIs and methods are stable – unlike in a legacy DXP, there is no need to update your installation and custom code as new features and functionalities are released.
2. The App Framework provides a consistent means of managing (and scripting) installations of custom functionalities, an advantage of being on a modern architecture.
3. Apps can be installed and tested in sandbox environments, making continuous integration far simpler and easier. Many legacy systems often lack a way to easily back out of installed custom functionalities, requiring completely separate installations for testing.

Gartner concludes their DXP report with a note: “Some DXPs come with capabilities for DevOps and continuous integration/continuous delivery (CI/CD) pipeline management to assist integration and deployment.” These capabilities are key to a modern, scalable platform and while some DXPs have managed to bolt them on after the fact (sometimes at extra cost), many others do not have them at all.

Conclusions

When evaluating digital experience delivery platforms, it's extremely tempting to get caught up on features, while not realizing that there has been a fundamental shift in how digital experiences are delivered. Agility is now a major factor in experience delivery and differentiation.

Another common pitfall when evaluating marketing technology platforms is overbuying. This causes a number of problems; first, overbuying introduces greater expectations on the platform, rather than focusing on internal team enablement. This leads to longer waterfall timelines (especially if you are trying to introduce new capabilities that didn't exist before) as well as a larger gap between expectations and delivery.

Second, the platform introduces complexity that didn't previously exist. Often, these additional layers (such as search technology, caching, CDNs, additional processing servers, etc.) are introduced to deal with scale, but just add unneeded functionality for smaller use cases. Overbuying is also simply a waste of budget and resources. Few organizations are in a position to overbuy based upon lofty growth goals, especially when cash flow is tight in the short term and quick wins are required.

It's far more effective to have a platform that scales with your needs. Most of Contentful's largest enterprise customers started small – in both team size and ambition. As they saw success, they started looking at larger digital experience projects – eventually scaling the platform to work across teams, channels and use cases.

As Forrester concludes, "Removing operational friction will be the hallmark of a successful Agile CMS strategy." Scalability in systems and processes is one of the largest sources of friction in any large organization. Rather than overbuying, it's important to have a platform which allows organizations to get started quickly and scale as they add new channels and business goals.

See how user friendly a content platform can be

[Talk to a Contentful expert today](#)

