Srijan Helps a Leading Publishing Conglomerate take Component Based Development Approach for Rapid Rollout of Brand Websites

CRAIN

SRIJAN

SUMMARY

Project URLs:

https://www.autonews.com/
https://mexico.autonews.com/
https://canada.autonews.com/
https://europe.autonews.com/
https://adage.com/creativity and https://adage.com/amp
https://chicagobusiness.com
https://www.crainsdetroit.com/
https://crainscleveland.com
https://crainsnewyork.com
https://modernhealthcare.com
https://www.plasticsnewseurope.com/
https://www.prw.com/
adage.com

The Client

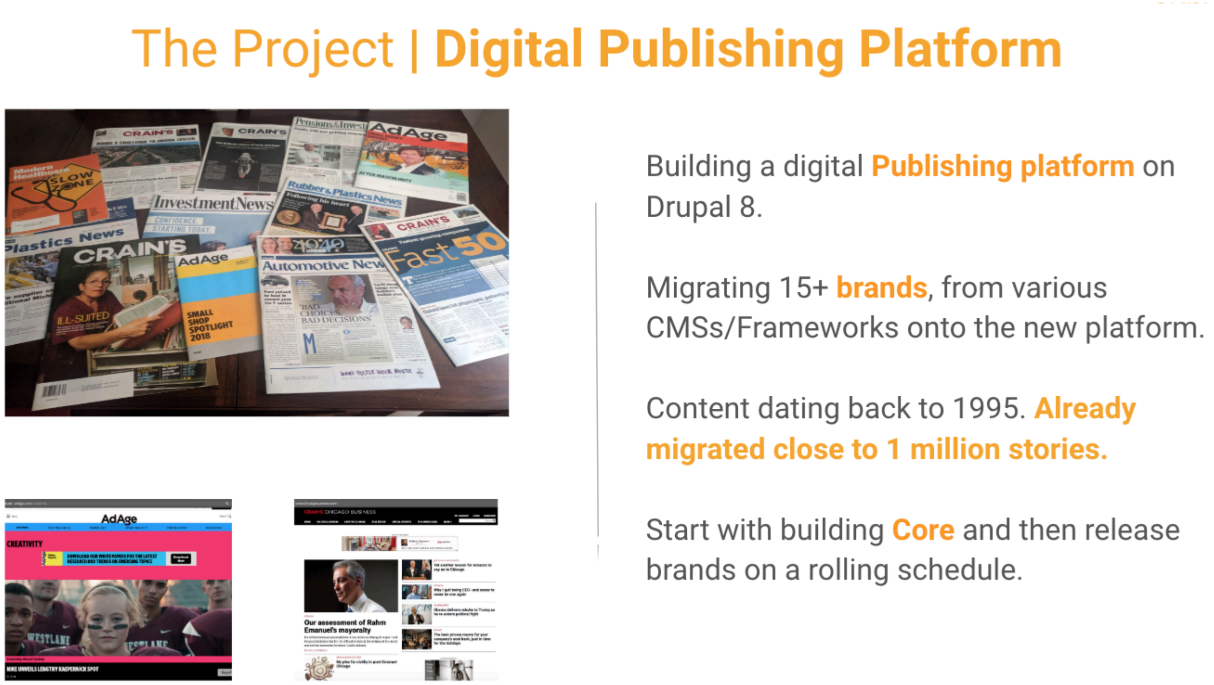
Crain Communications is one of the largest privately owned business media companies with 55 leading business, trade and consumer brands in North America, Europe and Asia.

Vision

As one company and one team deliver a unified, scalable and highly configurable digital platform that enables our business teams to quickly and effectively tell the stories that our customers value.

The Challenge

Each brand in the Crain family was operating on its own web platform. Rolling out a new digital initiative across all brands would require a lot of coordination and time. These disparate technologies systems also introduced operational inefficiencies. As there were no central governance place as all websites were working independent of each other, storage costs were rising, and sites were not managed efficiently.



The Project | **Digital Publishing Platform**

Building a digital **Publishing platform** on Drupal 8.

Migrating 15+ **brands**, from various CMSs/Frameworks onto the new platform.

Content dating back to 1995. **Already migrated close to 1 million stories.**

Start with building **Core** and then release brands on a rolling schedule.

The Solution

The approach was to leverage the Acquia **Lightning**, a distribution of Drupal 8 to enable component based development as well as deploy a Lightning sub profile on a multi-site architecture for all the brand sites.

Srijan proposed to standardize the components as the development progressed. The idea was also to bring better governance to have a tighter grip to the architecture and ensure a documented process approach towards components.

This is how the development was planned:

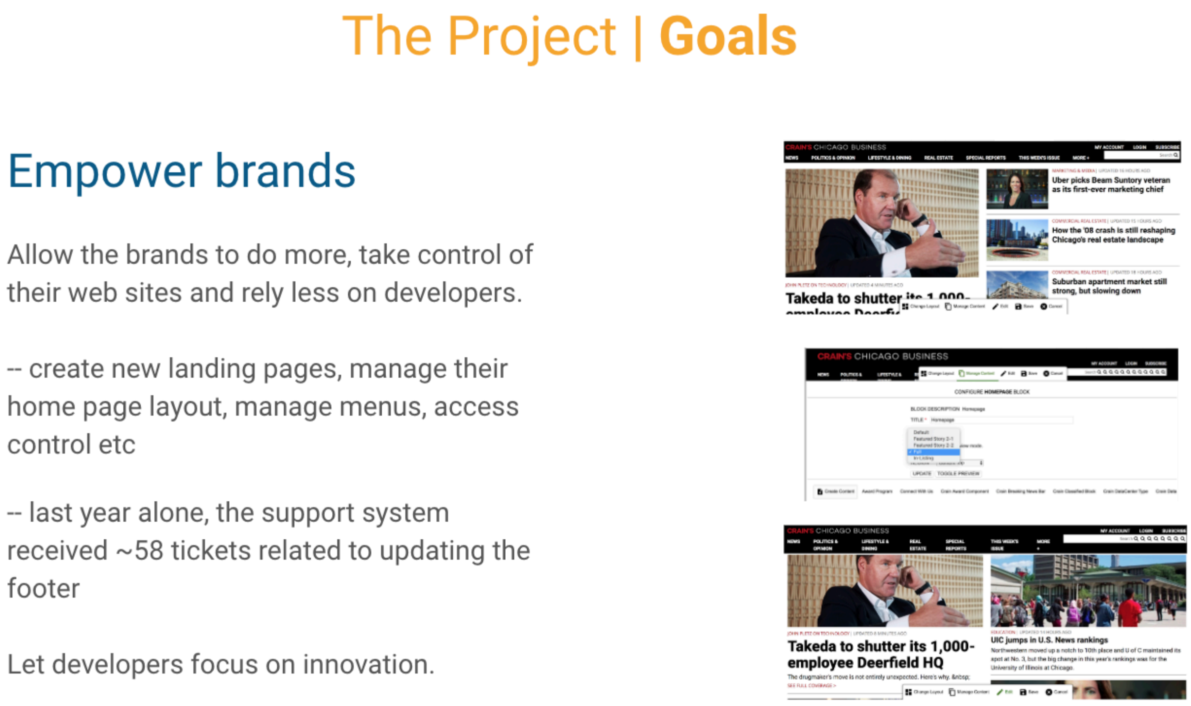
- Build out the core profile which encompasses most of the functionality, so the core could have 60%-80% of all brand features available out of the box.

- Each site to be then built uniquely with its own theme and UI components.
- Each component to have a set of configurations and view/display modes, allowing each brand to customize the site to their needs.

The idea was to empower the brand teams so that the majority of the site pages and features could be set up by them, without needing any development support.

Srijan used [Pattern Lab]() to codify the design guidelines for the components, which has now become the central style guide for the client's website design components.

[Click here to watch a small demo of the platform features]()



The Project | **Goals**

Empower brands

Allow the brands to do more, take control of their web sites and rely less on developers.

-- create new landing pages, manage their home page layout, manage menus, access control etc

-- last year alone, the support system received ~58 tickets related to updating the footer

Let developers focus on innovation.

CRAIN'S CHICAGO BUSINESS
Uber picks Beam Suntory veteran as its first-ever marketing chief
How the '08 crash is still reshaping Chicago's real estate landscape
Suburban apartment market still strong, but slowing down
Takeda to shutter its 1,000-employee Deerfield HQ
CONFIGURE HOMEPAGE BLOCK
UPDATE
TOGGLE PREVIEW
UIC jumps in U.S. News rankings

The Project | **Goals**

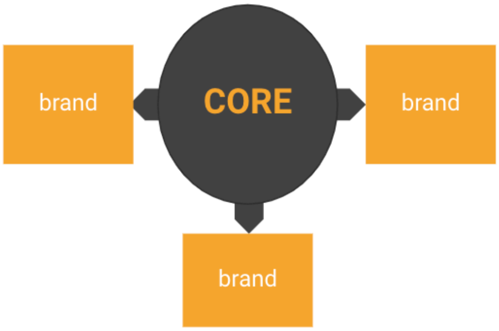
Develop once, use anywhere

All brands should reap the benefits of best in class features.

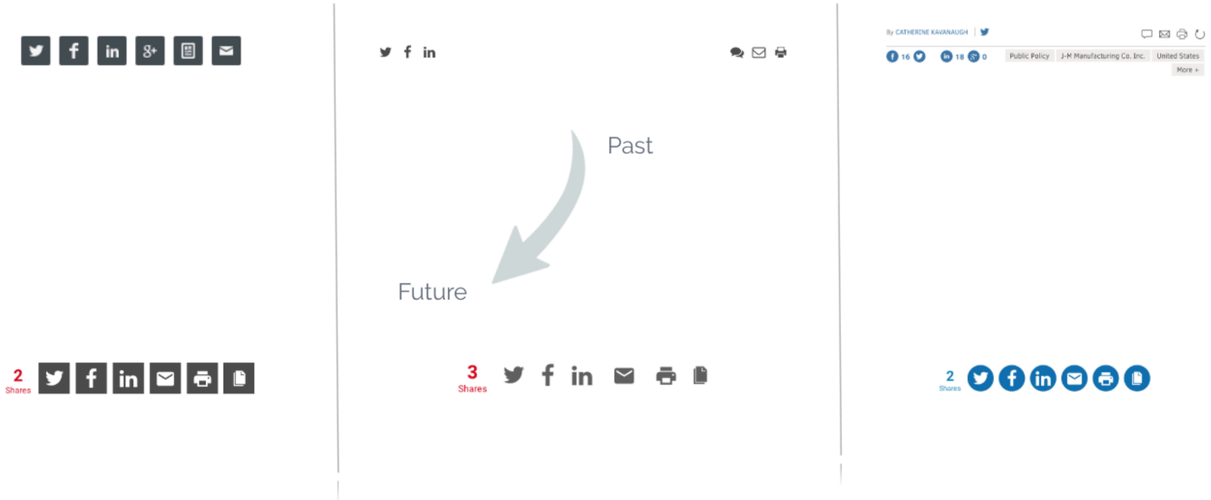
-- Brand X does something well, adapt it and distribute to others.

Reduce maintenance costs over time.

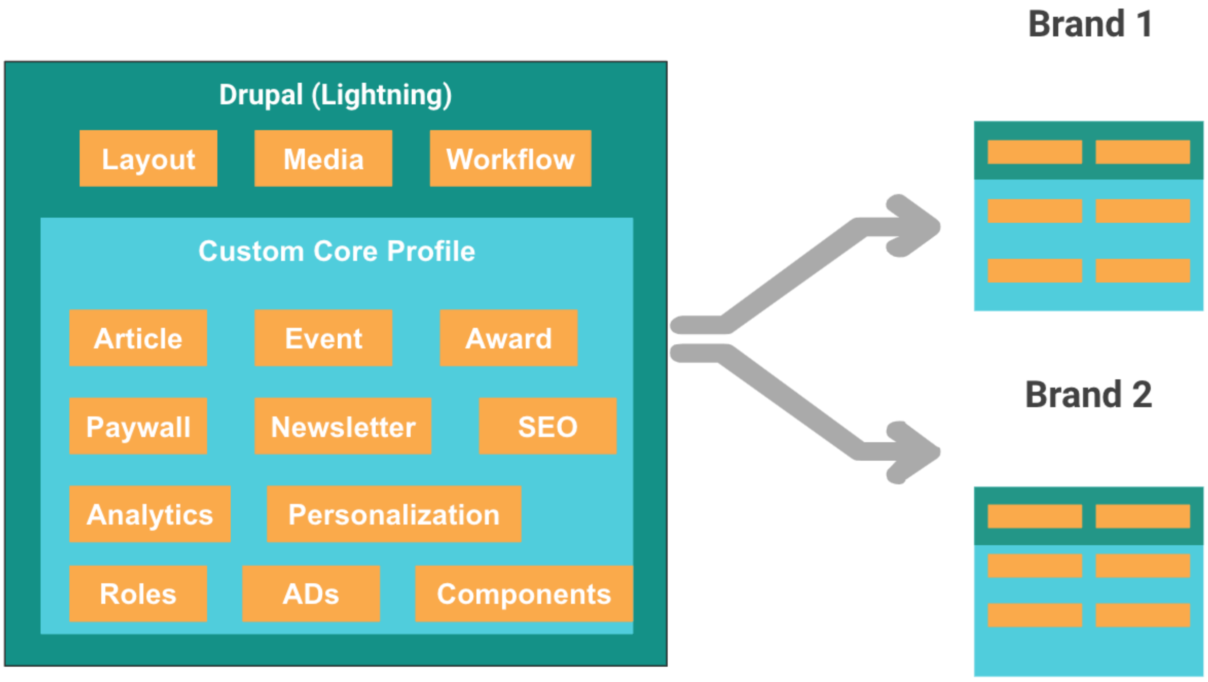
Maintain code standards and best practices (SEO, accessibility, analytics etc) across brands.



The Project | **Goals**



The Project | **Goals**



Brand 1

Brand 2

Drupal (Lightning)

Layout Media Workflow

Custom Core Profile

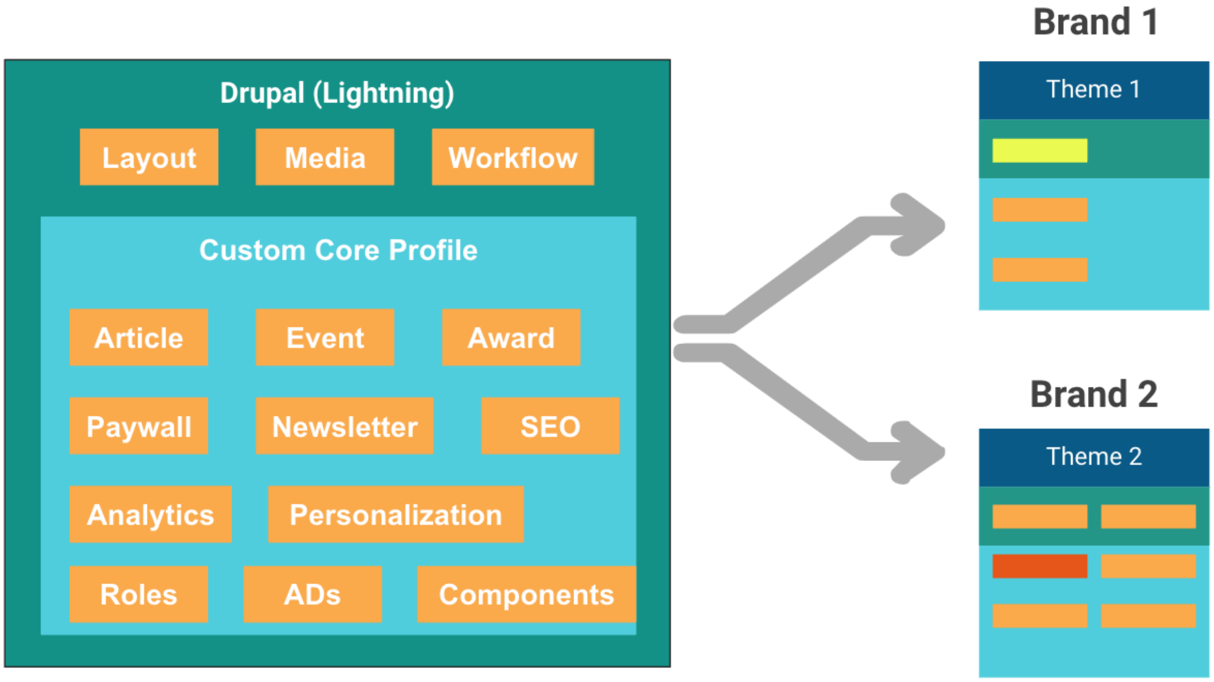
Article Event Award

Paywall Newsletter SEO

Analytics Personalization

Roles ADs Components

The Project | **Goals**



Brand 1

Theme 1

Brand 2

Theme 2

Drupal (Lightning)

Layout Media Workflow

Custom Core Profile

Article Event Award

Paywall Newsletter SEO

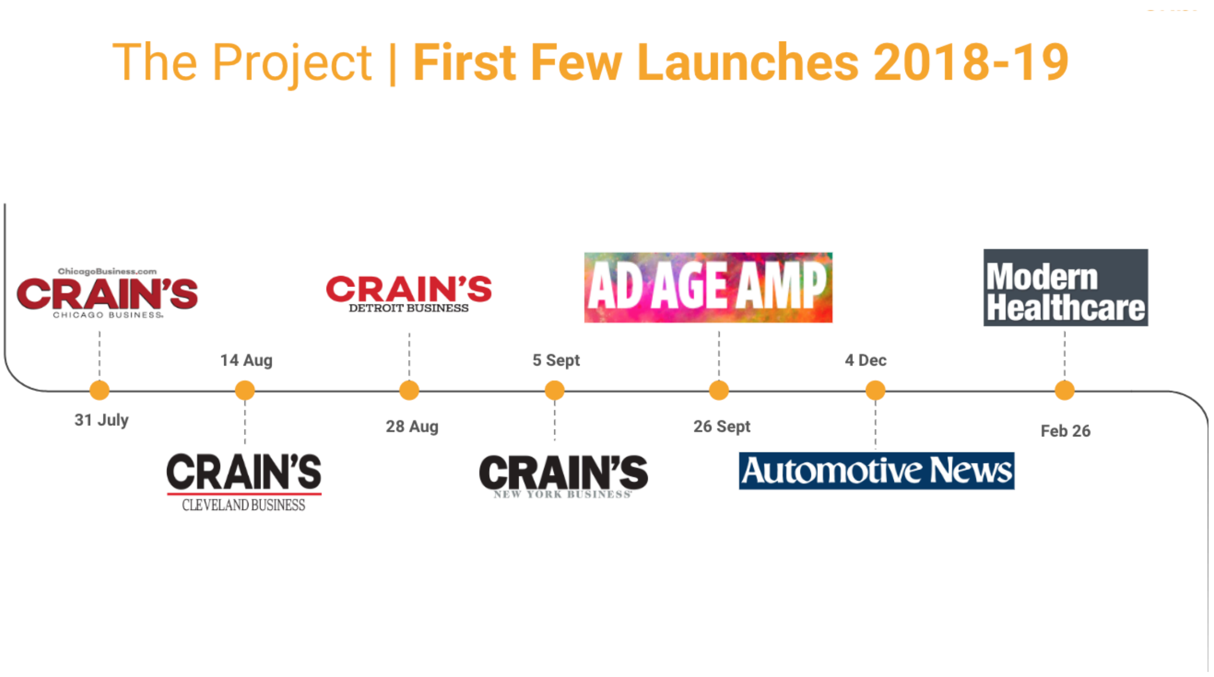
Analytics Personalization

Roles ADs Components

Click here to watch a small demo of the platform features

The Results

The core profile took nearly seven months to build. **The first four sites went live in August-September of 2018, within 6 weeks of each other.** Investing seven months on building the core platform paid out with the first four launches itself. Earlier, the estimate to redesign and migrate each web site (if done independently) was three to five months.
The return on investment increased as more brands are on boarded onto the platform.



The Project | **First Few Launches 2018-19**

Key results:

- Rapid brand rollouts
- Engineering team's time is freed up -the first brand was able to set up around **50 landing pages** without needing any developer intervention.
- All brands share best practices across analytics, SEO (meta tags, etc.), performance (Drupal Cache, Varnish, Cloudflare), media management etc as these are part of the Core platform.

The Development

Lightning comes with powerful out-of-the-box features such as the Layout, Media and Workflow capabilities. Srijan built a Crain core profile that packed the full set of features and functionalities already defined by the client's team for the brand websites. When a new website for a brand has to be built, the custom core delivers a functional website with most features already in place.

While the custom core was being developed, the client's design team worked on the new look and feel for the various brands as per the requirements and brand guidelines. These design templates were then given to the Srijan team. Using Pattern Lab, Srijan's front-end team built out the components used within the designs. A global style guide was created and is now maintained on Pattern Lab, which is now the central place to see all components within the platform as well as the specific ones used by the brands.

The client's product team and Srijan's development team worked with a list of components that would be common across all the websites. These were then brought into Drupal 8.

This also meant that the front-end team didn't have to wait for the backend team to get started on their work. They could independently start coding out the components without having to worry about the backend capabilities.

Drupal 8 lends itself very well to the component based approach, adding far more site building power than the previous versions of Drupal. Here are the modules we used.

DRUPAL 8 MODULES
Panels and Panelizer
Layout Discovery
Blocks as Entities
Views
Paragraphs and Quick Edit
Media Entity

Srijan worked on a component approach at two levels.

- The Block entity defines how independent, non structured content works. For example, a component that shows a listing of articles, or a Twitter feed.

- The Content entity defines how structured content like an article or a story works. So instead of building an article entity as a single monolithic entity, we divided the content entities into components using the Paragraph or Media Entity modules. Dividing the content entities into sub entities not only gives a better editorial experience but also increases re-usability of the sub-components. For example, if we show a video/image slider within an article the slider can be a component built with the Paragraph or Media Entity module.

Once the core profile was readied, the brand websites could now be built up fairly quickly. An instance of a brand website was built up from that core with its own theming. Then it was time for migration of the website's old content to the new one. Brands validated the migrated content, set up landing pages and now do other site building activity without the intervention of developers.

We leveraged Acquia's enterprise cloud solutions to support a large multi site setup. The combined unique site visitor traffic to the client's websites is pretty significant and the shared multisite setup means that we needed the ability to scale up on demand, and as new brands go live.

We also used Acquia BLT, the build and launch tool, that provides an automation layer for testing, building, and launching D8 applications. With 30+ developers working across timezones, multiple tracks running in parallel (live sites + under development sites on the same shared core codebase), continuous integration and continuous development was key. We used BLT extensively to set up our development and deployment pipelines.

Empowering Editors

This is how the new component based development is helping editors at the client's websites to have full control on how they can work with content and layouts.

Editors can choose a layout for a new page they want to create. And choose the components from the library that is available. These components could be an article with a thumbnail, or a list of articles with an featured article and image, or a promo box, or a panel of social media icons to enable sharing. All these have been built using the Blocks as Entities module, which pulls the template information from the Pattern Library Srijan created. Blocks are entities with fields in Drupal 8, and they serve as the backbone for the component backend.

Each component that an editor chooses for the page is editable from within the layout view itself. So there is no complex backend system where the layout is being created for which a view action is then required. The editor gets the editable page as it will render on the website, and gets into the edit mode right from that layout. So editors get a consistent editorial experience.

Each component can be set up with different view modes, so you can have the same component being rendered in different modes, for example, as part of a list, or as a full element, and so on.

What does all this mean? The layouts for pages are already defined. So are the components and the various view modes for them. For example, a component that shows the top 10 articles can be rendered as a listing or as a grid view. So a block entity serves as a re-usable backend entity for both the view modes on which we can define two different views modes. Editors can select the mode while adding/updating the components on a page.

For each site, the components and layouts are defined in their brand colors and guidelines. So while the base components developed are the same across all the sites, how they appear on a particular website is based on the theme of the website. The editors get a predefined canvas to play around, but also know what can or cannot happen on the pages.

What about content and tags and categories? And archived stuff? All that’s easily editable as well. The Blocks as Entities module essentially gives developers to build components to which various fields can be added. So each component can have different fields and capabilities, which delivers another level of freedom to the content editors. They can add articles, move them around, move them up and down, and so on. Components also can be built to automatically pull in articles or content pieces that belong to a specific category or have specific tags., using Taxonomy terms. Other ideas for components include where different types of content types can be pulled in, not just articles, but videos and photo galleries as well.

This essentially means a content editor now has the full flexibility to choose not just how the content should be presented, but also have a format to pull in relevant content automatically. Which means while they use a highly useable field format to choose their options, the D8 backend is running a smart query to bring them the information they want to render on the site. It may not sound much like to a small website with limited content. But for a media company that went digital in the nineties, that’s gigabytes of content that needs to be around and used when needed.

With the vastly improved Paragraph module in Drupal 8, content editors can add a host of content types inline very easily, and edit each element right there. Adding related content inline can be a messy task, as most content authors know, and they often have to rely on the development team to get the layout fixed.

Migration

Migration of data was a big challenge, as the client’s websites have data going back to the mid-90s. In the interest of saving of time, the migration was planned in such a way that training of editors on the new Drupal 8 platform happened at the same time, without impacting the experience of the editors in any way.

Srijan exported the data from the proprietary platforms using XML and CSV formats. For migration from Drupal 7 to D8, a custom module was written.

Image Optimization: We found that high resolution image use was widespread on the websites, as the existing CMSs were not set up to auto optimize the images. So in the migration process, we used the GD2 image manipulation toolkit to optimize image size without degrading the image quality while also considering retina displays.

De-Duplications: There was no set process for using images and multimedia files. This led to the same images being uploaded multiple times to the CMS, often with different names, so it was challenging to even identify them. Storage needs as well as costs were increasing.

Srijan did a deduplication effort for this. Instead of identifying an image with its name, it is now identified in the binary form. Only unique images and files were retained in the system. Now if an editor attempts a re-upload of an image, the system will alert her about it and she can use the existing one for her story.

Since reducing costs was a major goal for the client’s team, we also moved their storage setup to Amazon S3 from their existing setup.

Content categorization: Over time the websites saw a huge number of categories being created, even though they already existed in the CMS. This impacts the quality of search results as well as creates confusion for editors. Srijan reviewed the categories along with the client’s business team and decided upon the unique set of categories to retain, and migrated the legacy articles with the new categorization rules.

Migration and training, simultaneously: Since the project had strict deadlines, the migration process had to happen in hand with the training of editors on the new platform. This essentially meant that any migration work that happened was not to have any sort of impact on the user experience of the editors who were learning to use the new website before the go-live. With every phase of migration, we had to ensure that the unique content IDs for the content being pulled into the components remained the same. This was done using the Migrate module.

Reusable migration code across brands: Srijan's migration team followed the migration strategy recommended for D8, and the code was developed as re-usable plugins to process the data. These re-usable plugins were developed as part of the core profile of the client, which was extended/overridden at the brand level to satisfy brand specific use-cases. This approach helped the migration team in running the migration of multiple brands in parallel with less and reusable code, and better efficiency.