Component-Based Design with Acquia Site Studio

Streamline the Process of Building Websites and Webpages

Table of Contents

What Is Component-Based Design?

Component-based design systems have evolved quickly over the past few years.

They're now considered by designers, developers, product owners, and project managers as the most effective way to develop and manage design, especially at scale.

In this e-book, we will show you how a component-based design system can be implemented within Drupal using Acquia Site Studio.

Component-Based Design

There are many names used for component-based design including pattern library, design system, living style guide, and atomic design.

Whatever you call it, the idea is that rather than designing page by page, you break a design down into smaller component parts. Individually these can be quite simple, but when combined together they create something more meaningful and useful. And when combined and arranged in a different way, their meaning and use can change.

At the heart of component design is the philosophy: **Create once, use many.**

Benefits of component-based design systems include:

- **Better communication through a common visual language**
  Component design systems allow for a shared general knowledge of the components available to designers, developers, and project managers.

- **Consistent design and UX**
  Designs, which are formed from a common set of components, create a more consistent "look and feel" and user experience.

- **The ability to create new digital products faster**
  Component design systems minimize the time spent redesigning and rebuilding similar components. As mentioned above, this can be incredibly important for organizations that need to scale up websites and webpages quickly.

Approaches to Creating a Component-Based Design System

Designers and developers tend to apply many different approaches to creating component-based design systems. **It isn't always a simple task, though.**

Challenges can include:

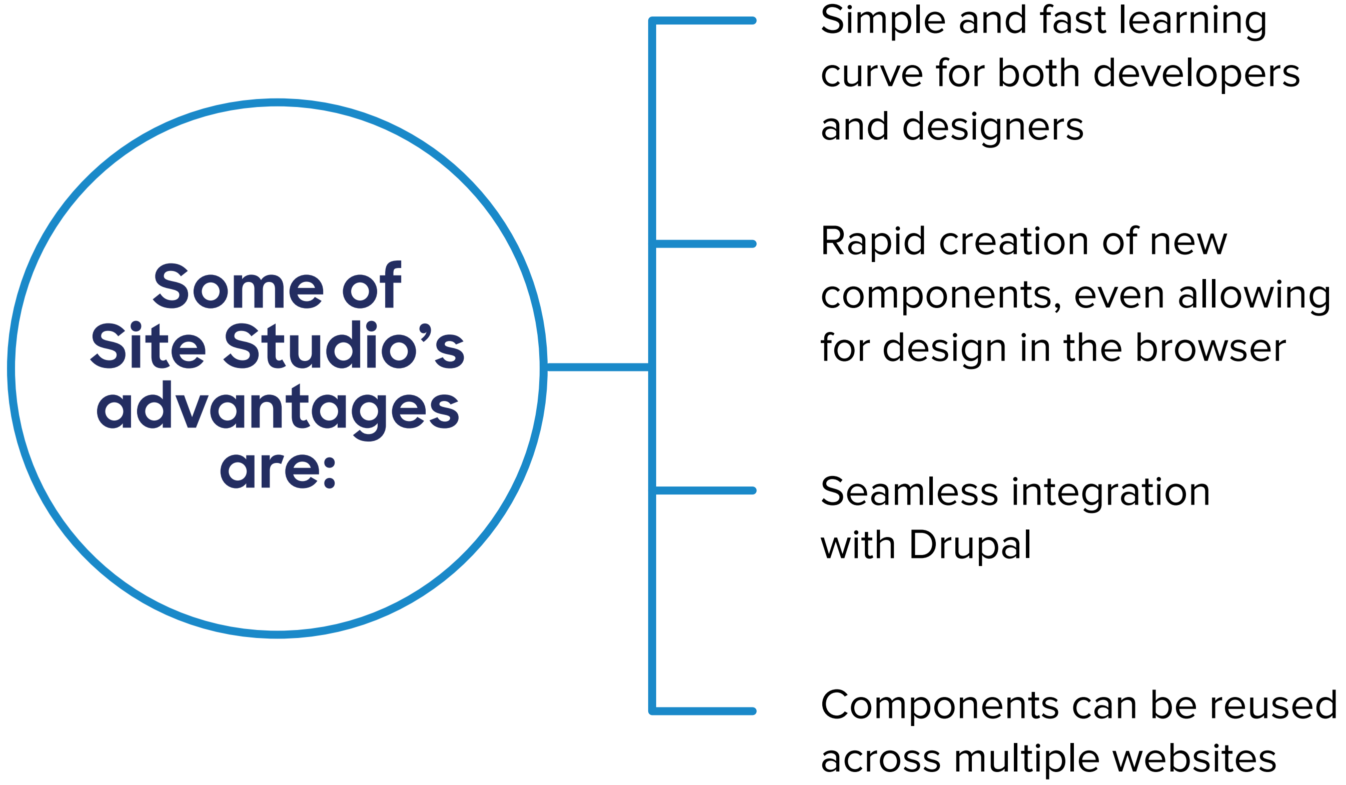
- Complexity of design
- Technical implementation
- Organizational buy-in
- Ongoing management

For large organizations who have a lot to gain, it's worth the effort and cost. For these, extremely technical React-based implementations are an attractive option. Some companies, like Airbnb, have even gone as far as integrating their code-based design systems with design packages like Sketch.

For organizations that don't have the technical resources (and budgets) to build such technical systems and digital agencies who want to create component-based design systems for a broad range of clients, it's not always possible or cost-effective to implement such solutions. This doesn't, however, reduce the need or advantage of using them.

Creating a Component-Based System Within Drupal Using Acquia Site Studio

Site Studio provides developers and designers with a powerful set of visual UI's within Drupal, which will allow for the creation of a component-based design system or pattern library. It also provides website editors and content creators with an intuitive interface for creating new content from components from the library.



Some of Site Studio's advantages are:

- Simple and fast learning curve for both developers and designers
- Rapid creation of new components, even allowing for design in the browser
- Seamless integration with Drupal
- Components can be reused across multiple websites

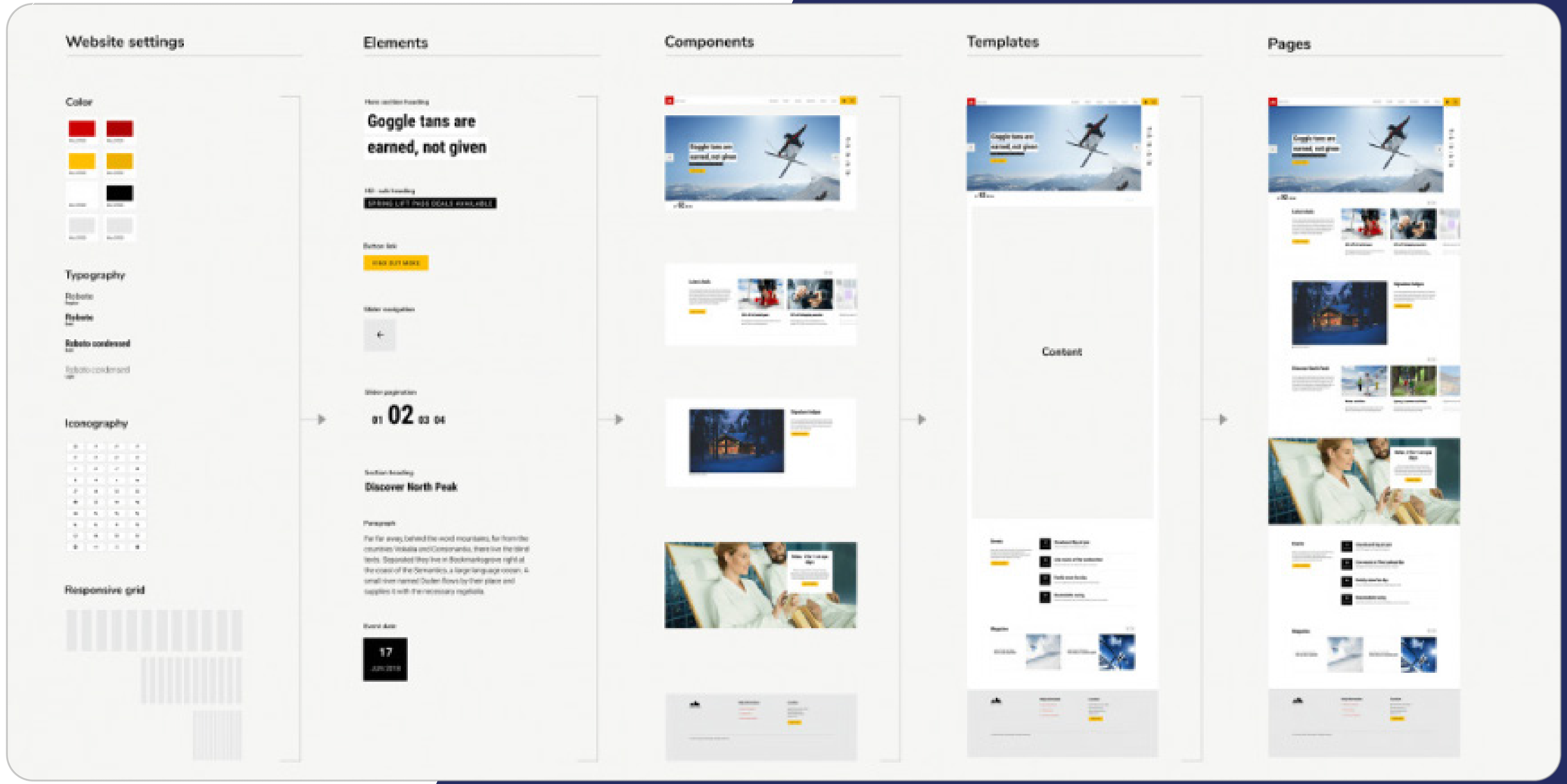
Component-Based Design with Acquia Site Studio

How To Create with Site Studio

What Can You Create Using Site Studio?

Site Studio allows you to create many parts of a website that can be considered conceptually as components within a design system. These are illustrated in the diagram to the right. As you can see, you can configure the basic website settings (color, fonts, icons, and grid) and create element styles, layout and content components, templates, and page layouts.

All of these can be considered components of a design system but as you may have noticed, one of the columns is actually labeled "Components." It's these that we are going to focus on because they are the most literal in terms of component-based design.



Components of a design system

What Are Components?

Put simply, components are templates that contain the layout, style, functionality, and editing experience they need to be used. The concept is not unlike the Paragraphs feature in Drupal. Though components are primarily meant to serve as static building blocks rather than functional elements, you can create components with functionality built into them. Technically, Site Studio components are Drupal configuration entities that when passed via API to a compiler, generate a Twig file, CSS, JS, and JSON for the form settings.

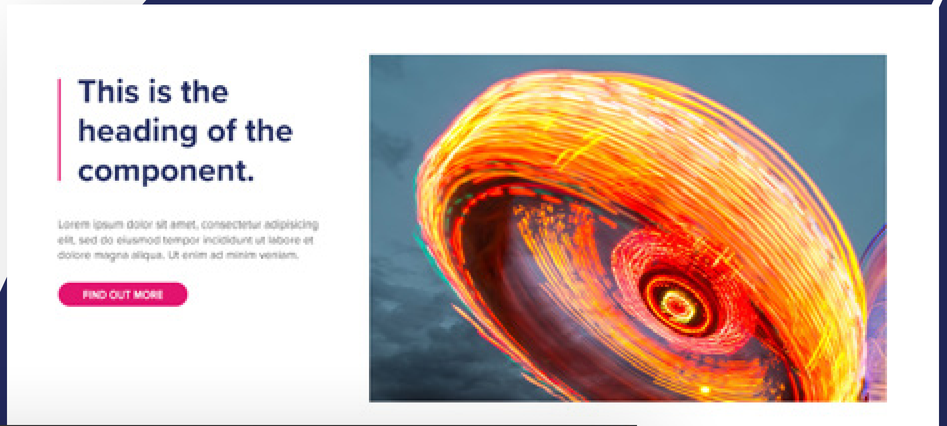
Site Studio components can be used by editors within pages to create component-based layouts and content.

They can also be used by site-builders within templates to take a component-based approach to the complete website build. To the right is an example of a simple page section component that would be used by a website editor. It includes a layout with two columns, a heading, a paragraph, a link, and an image.

Below that is the custom form that users of the component will use to edit its content and settings.

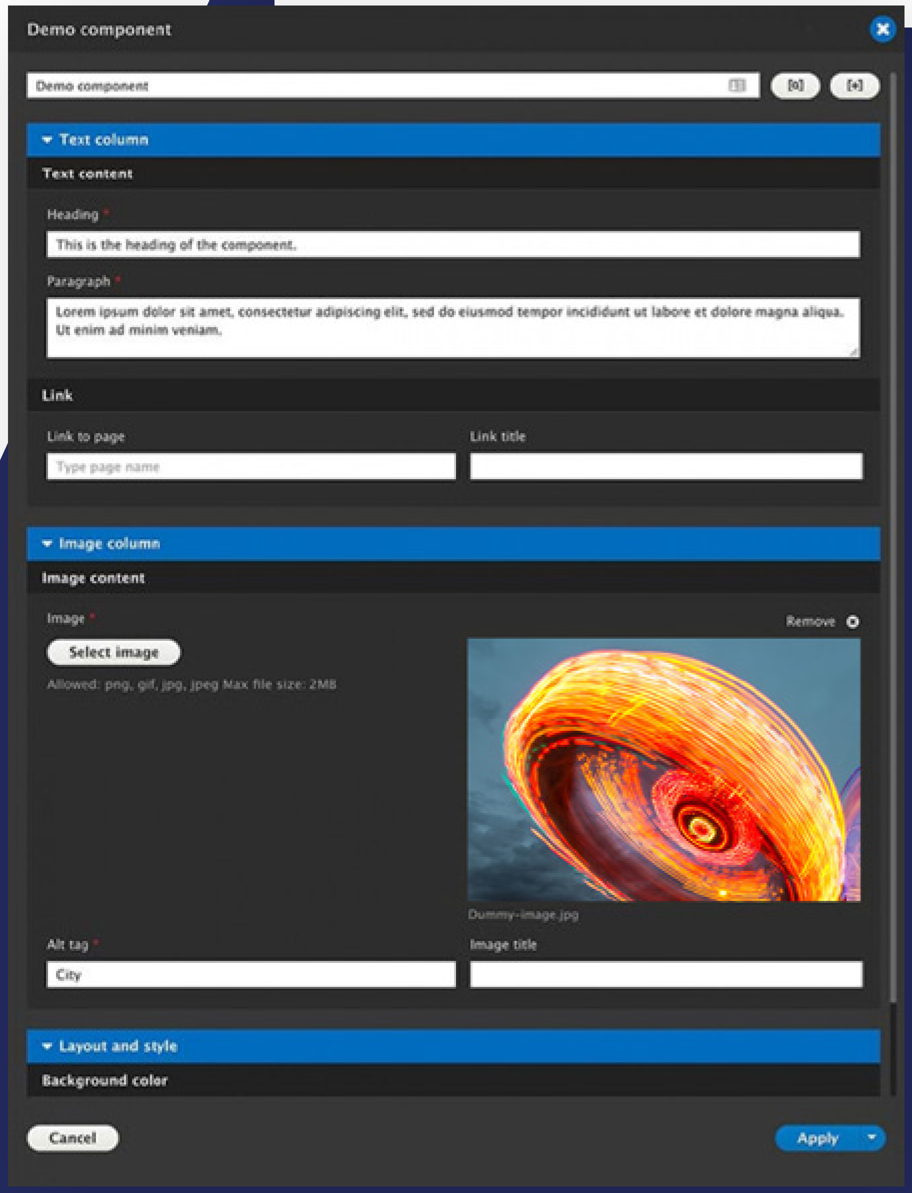
We're going to walk through the process of creating a component but before we do, there are some component design considerations that are useful to understand first.

Simple component design



This is the heading of the component.

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam.



Demo component

Demo component

Text column

Text content

Heading *

This is the heading of the component.

Paragraph *

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam.

Link

Link to page: Type page name

Link title:

Image column

Image content

Image *

Allowed: png, gif, jpg, jpeg Max file size: 2MB

Dummy-image.jpg

Alt tag *: City

Image title:

Layout and style

Background color

Site Studio component editing form

Keep These Component Design Guidelines In Mind

Design lean, build early.
Designing every page of a website at multiple breakpoints and then handing it over to a developer will always lead to a costly back-and-forth exchange. Ideally, designers, developers, product owners, and other stakeholders work together to define the initial list of components required.

Keep the first set of component designs to a minimum before you move to building them.

You can gradually build up a library of components as you need them. This way, your learnings from building the first set can be applied when designing any additional components. It will also help to keep your library bloat-free.

It's also a great time to involve external stakeholders. If you're in an agency, client feedback and real-world testing with actual content can be done at this stage. The goal is to avoid large-scale redesigns.

Create once, reuse many.
This is probably the most important rule and the easiest to break. You have to be quite ruthless when managing a component design system.

Ask yourself the question, "Will creating a variant of an existing component really help the website so much that it's worth increasing the size of your component library?"

From our experience, the larger the library, the more difficult it is to manage, and the larger the risk of design creep and inconsistency.

Try to be efficient with your component design. Sometimes a single component with some basic options for configuration will be more useful than five separate components.

Design in the browser.

Design systems evolve; it's inevitable and actually desirable. You want them to keep pace with changing user requirements and expectations. If you have a large design and engineering team, you may be able to keep your design systems in sync between code and a design package like Sketch.

If this becomes too difficult or costly, then your "built" components should be the single source of truth. Evolving a component in the browser will ensure that website users are using the most up-to-date design.

Use what you've created.

We've seen design and development teams create component-based design systems only to then have designers turn back to their favorite design package to lay out a new page or interface. This can be a real waste of time and budget.

If you've created a set of components with a commonly understood naming system and knowledge of what they do, you should be able to sketch out new user interfaces on paper, whiteboards, or even the back of your hand. Designers, developers, and content creators should be able to create new layouts easily from the library. If they can't, something isn't set up correctly.

Pro Tip

Steer Clear of These Component-Related Pitfalls

RESOURCE CARD	BUTTON	FORM	LIGHTBOX
RESOURCE CARD A	BUTTON A	FORM A	LIGHTBOX A
RESOURCE CARD B	BUTTON B	FORM B	LIGHTBOX B
RESOURCE CARD C	BUTTON C	FORM C	LIGHTBOX C
RESOURCE CARD E	BUTTON E	FORM E	LIGHTBOX D

Avoid Creating Supercomponents
Once you see how easy it is to create a component, you might be tempted to add numerous options, including potential functionalities, so that that component can be used to create many different webpage items. However, you want to strike the right balance between useful and simple. As you gain experience using Site Studio for your organization's websites, you'll develop a better sense of what options are most appropriate for each different component your team will use in their work.

Try to Prevent Library Bloat
Once you've got the hang of creating useful components, you might find it hard to stop. You might be tempted to save just about every component you create into your library. Though that might seem like a time-saver at first, it may make it more difficult to find just the right component you need later, and it can introduce inconsistencies between components more easily. It might take some practice to develop components that strike the right balance of utility, time-savings, and consistency. But once you gain more confidence there, you can build smaller, neater libraries that help expedite page building for everyone on your team who will use them.

Component-Based Design with Acquia Site Studio

Designing Components FAQs

What To Consider When Designing Components

Site Studio provides lots of options and opportunities when creating components. Although very useful, they also raise questions to think through:

What context will the component be used in?
Are they for developers to use within templates or are they for website editors to use for page creation?

How will components be used together?
You want to ensure that when components are combined and nested, they still appear and function as desired within your design system.

What is the skill level of the component user?
If your components are for editors, they should be easy to use. But at the same time, you want to provide them with the tools they want to create rich and engaging content. Sometimes this may include configuration options within the component that change its appearance and behavior. The skills of the component user may affect the complexity of the configuration options you offer.

What does Site Studio give you out of the box?
Acquia Site Studio includes a library of elements you can use to create components that range from basic HTML elements to sophisticated interactive devices. Although you can easily extend Site Studio by developing your own elements, it's more efficient to understand and utilize what's already available. This way, Site Studio keeps everything up-to-date and there's less of your own customized code to manage.

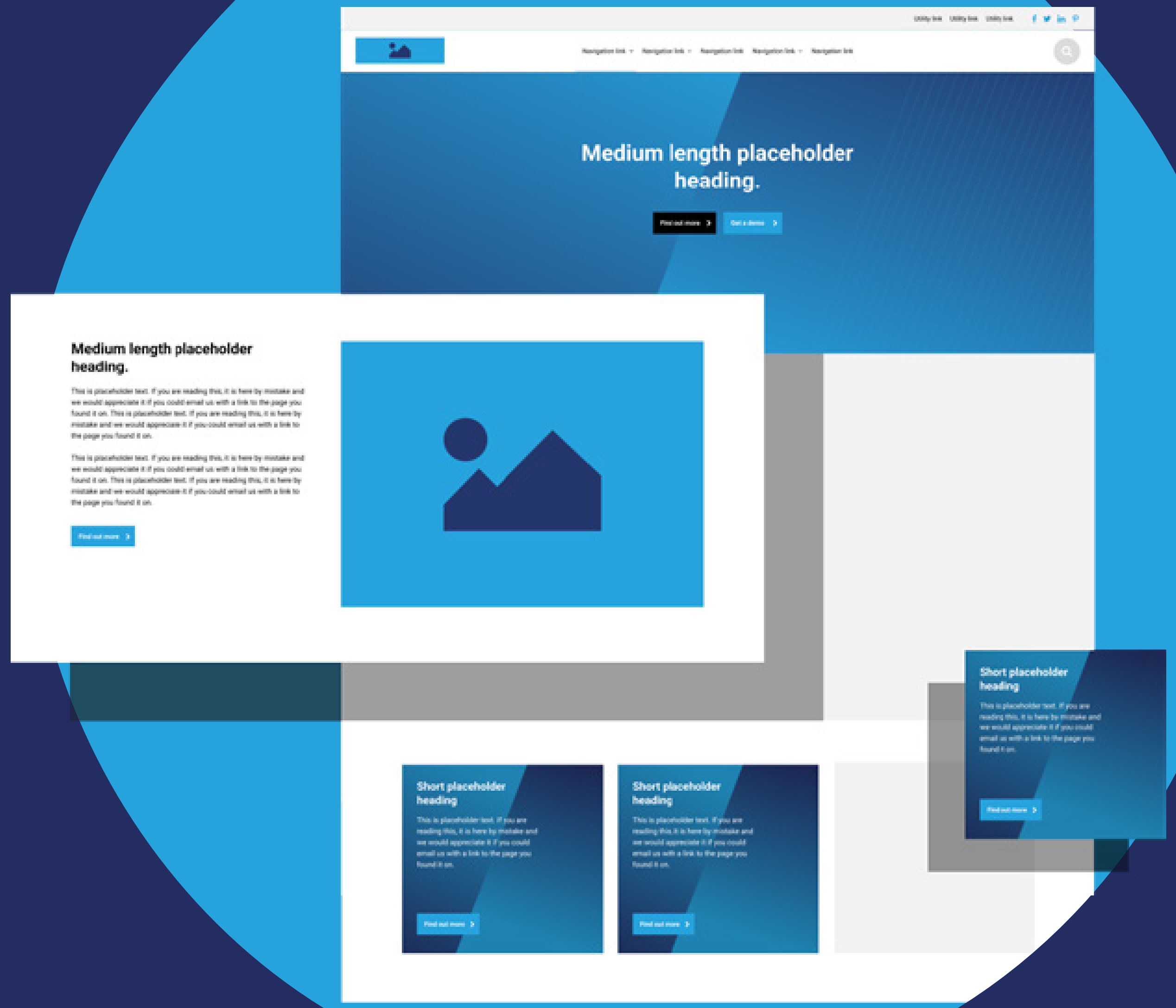
Answers to these kinds of questions will help you develop guidelines or specifications for component design and development. **Remember, the idea is to create consistency and this comes from a systematic approach.**

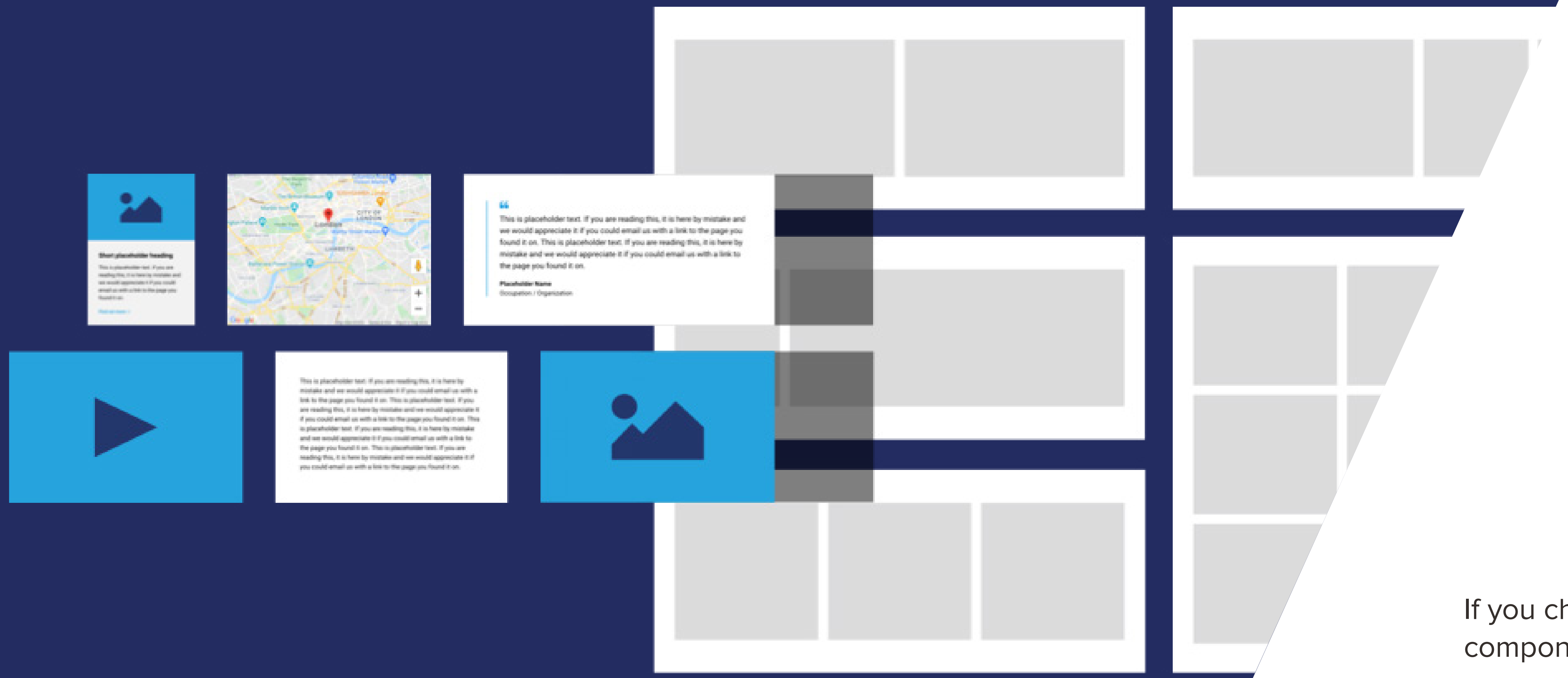
Pro Tip

Learn How to Use Components Quickly Through Site Studio UIKits

Ultimately, most designers and developers will want to create their own components, customized to the specifications of whatever website or digital project they are working on, and then save them into their own libraries. But Site Studio also gives users access to two UIKits that can help them get started with components quickly.

Sections UIKit
The Sections UIKit is a library of page section components that can be stacked, one upon another, to create pages that follow a consistent design system. Though this approach sacrifices some flexibility, the components deliver more design consistency and are probably a bit easier for content creators to use at first.


Medium length placeholder heading.
Medium length placeholder heading.
Short placeholder heading
Short placeholder heading
Short placeholder heading



This is placeholder text. If you are reading this, it is here by mistake and we would appreciate it if you could email us with a link to the page you found it on.

This is placeholder text. If you are reading this, it is here by mistake and we would appreciate it if you could email us with a link to the page you found it on. This is placeholder text. If you are reading this, it is here by mistake and we would appreciate it if you could email us with a link to the page you found it on.

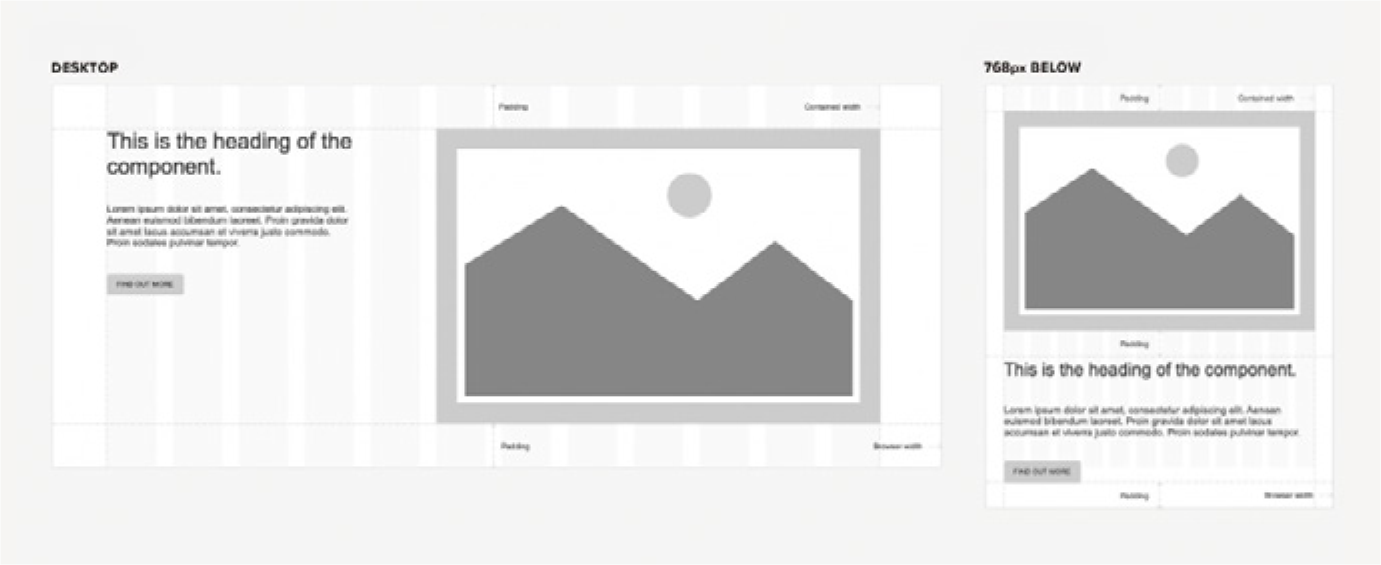
Placeholder Name
Occupation / Organization

This is placeholder text. If you are reading this, it is here by mistake and we would appreciate it if you could email us with a link to the page you found it on. This is placeholder text. If you are reading this, it is here by mistake and we would appreciate it if you could email us with a link to the page you found it on. This is placeholder text. If you are reading this, it is here by mistake and we would appreciate it if you could email us with a link to the page you found it on. This is placeholder text. If you are reading this, it is here by mistake and we would appreciate it if you could email us with a link to the page you found it on.

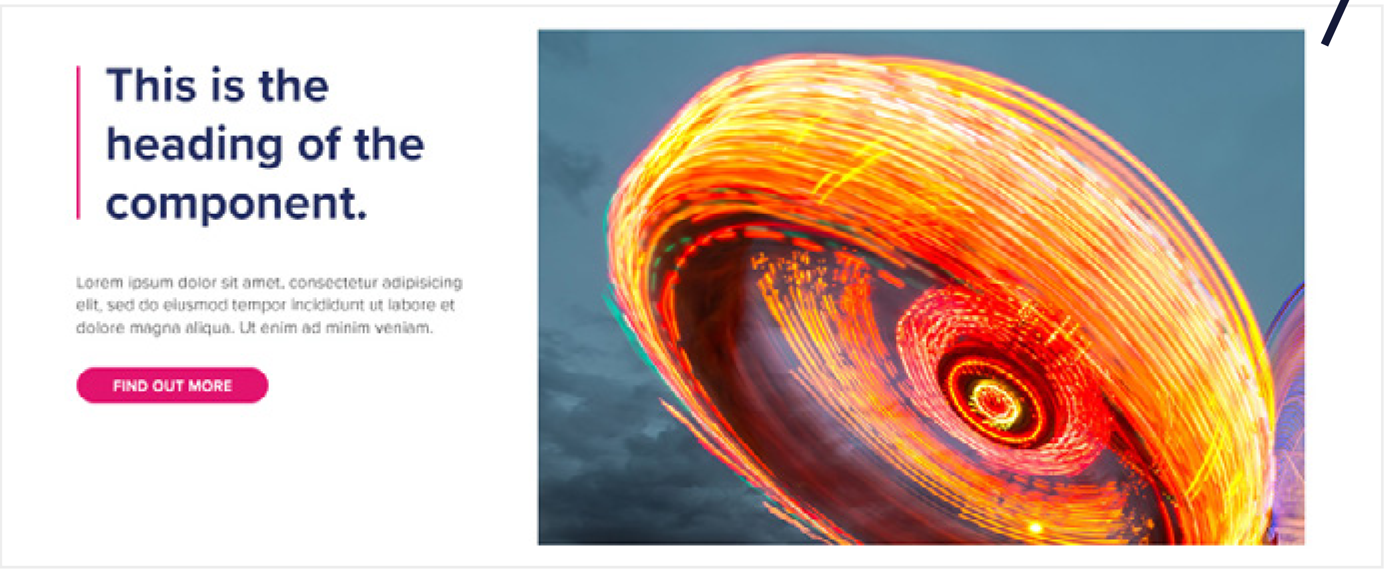
Primitives UIKit
The Primitives UIKit is a library of Layout components for your page structure and Basic components to add content. Though these components don't deliver the design consistency or ease-of-use that Sections components do, content creators have more flexibility. By combining these components, users with layout skills can create a huge variety of page designs without needing additional components.

If you choose to expedite your component-based design learning process by experimenting with a UIKit, we recommend that you stick with one approach rather than to combine them. Trying to combine different kinds of components like that can be confusing and lead to display issues. But broadly speaking, starting off with one of these libraries is a great way to become more familiar with components and to start building your own library of components.

Example of the Component Design and Build Process



Component wireframe


This is the heading of the component.

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam.

Component visual design

Basic Steps To Create a Simple Component in Site Studio

Step 1

Define the Component Requirements

The first step is to define the component requirements. In this example, the component requirements include separate fields for each content element to keep strict control over the content being added to the component. The component layout needs to stack at resolutions below 768px.

Step 2

Design the Component Style and Layout

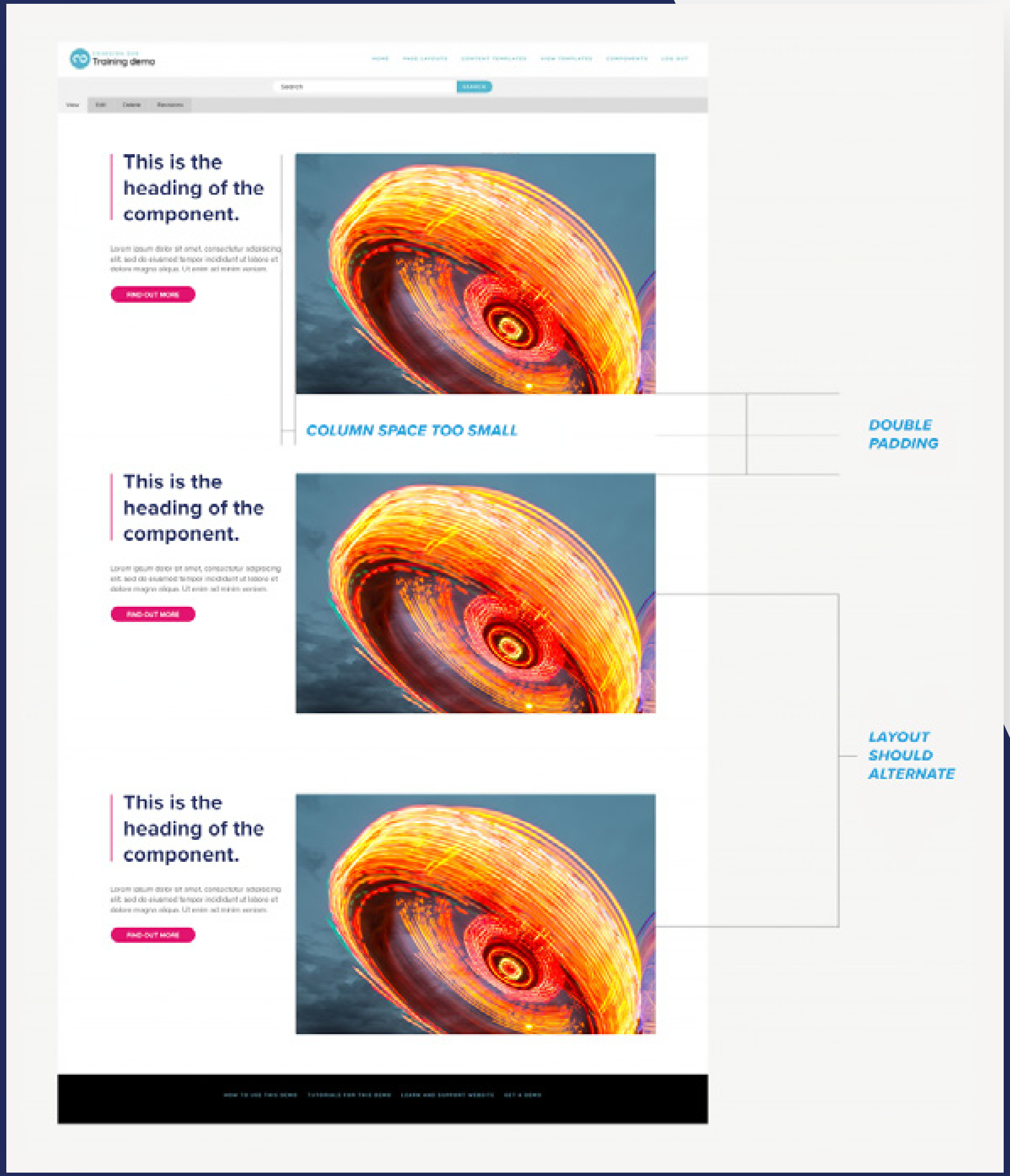
This can be done in a design package like Sketch or Figma, or directly in Site Studio if you want to design in the browser. Your choice depends on whether you're designing a component for a new website where the other parts of the design system (colors, fonts, styles) aren't in place, or adding a component to an existing website where they are. In this example, a Sketch design was produced first.

Step 3

Build the Component Layout

Using the Component builder, you can build the layout of the component. This should be done in a development environment rather than a production environment as you're creating a configuration entity, not a content entity.

As you are building the layout of your component using the drag-and-drop elements, you can preview what the component looks like using the preview. It's a good idea to add some dummy content to the elements so that you can see how it will look when populated.



This is the heading of the component.

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam.

This is the heading of the component.

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam.

This is the heading of the component.

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam.

Layout issues with component

Step 4

Test the Component Layout on a Page

Once the static component layout has been created, it's a good idea to test it. In this example, the component is a page section so it's important that it looks correct when used along with other components. The easiest way to test this is to add some existing components to your page and place them on either side of your new component.

It's also important that the component looks right when duplicated and stacked on top of one another. You can test this easily using Site Studio's duplicate function on the outermost element.

After running these tests, the following issues were identified when the component was duplicated:

- The top and bottom padding is doubled up and looks incorrect
- The space between the text and the image is too small
- The layout would look more interesting if the image and text position alternated.

Step 5

Evolve the Component Layout

Once any issues with the component layout have been identified, it’s a straightforward process to modify the layout using the layout canvas in the component builder.

For the issues identified in this example, additional space was added between the elements. Also, an alternating light gray background color and image position was implemented. It was also decided that the background color and layout change would be a configuration option in the component so that editors had some control over its layout and style.

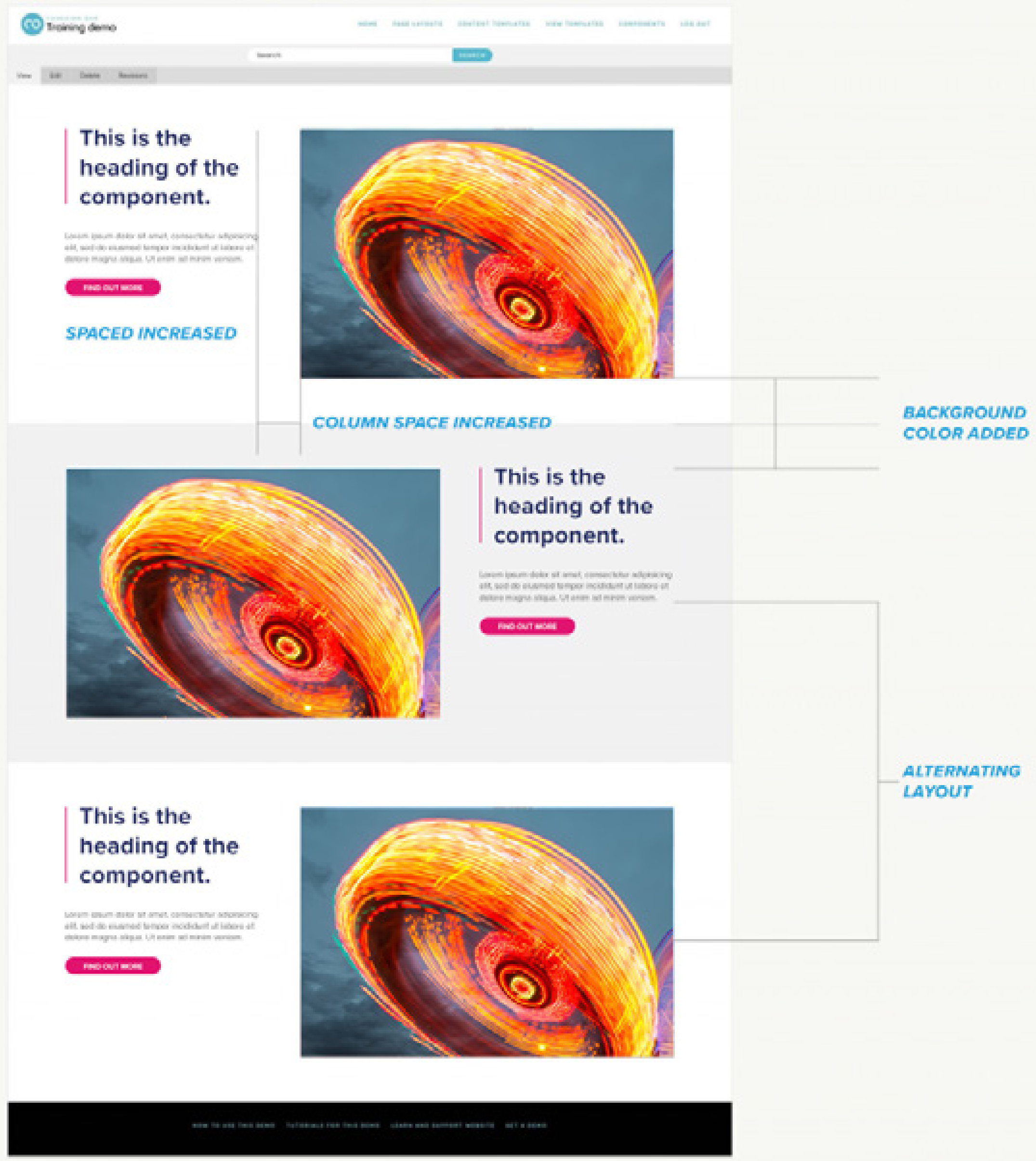
Step 6

Build the Editing Form and Test it on a Page

The next step is to build the editing form that content editors will use to edit the components content and apply any visual styling. In this example, the fields we added to the editing form for the content were: heading, paragraph, link label, link, image, image alt, and image title. We included default content and validation on the fields so that when the component is used on a page, it renders immediately, without the need for adding content. This helps the website editor construct their page layout quickly before editing the text and images.

We also added a drop-down field that will allow the editor to choose the position of the image and another drop-down field that will allow them to choose a color theme for the component. Once the component is complete, it can be tested by adding it to a page alongside other components and testing the editing form to ensure all the content is being rendered as expected.

Component layout issues resolved



This is the heading of the component.

This is the heading of the component.

This is the heading of the component.

Step 7

Deploy the Component to Your Staging or Production Environment

The component created is a configuration entity so it can be deployed as configuration. This can be achieved using Site Studio Sync. Site Studio Sync works the same way as Drupal's configuration export except that it looks for all the dependencies a Site Studio configuration has (including files) and adds them to a package that can be deployed to your staging or production environment. This feature is available via the command line (Drush) and through the UI.

Creating a Reusable Component Library to Use Across Multiple Projects

Once you've created a component (or template, style, or website setting), it's possible to export it from one website and import it into another using Site Studio Sync.

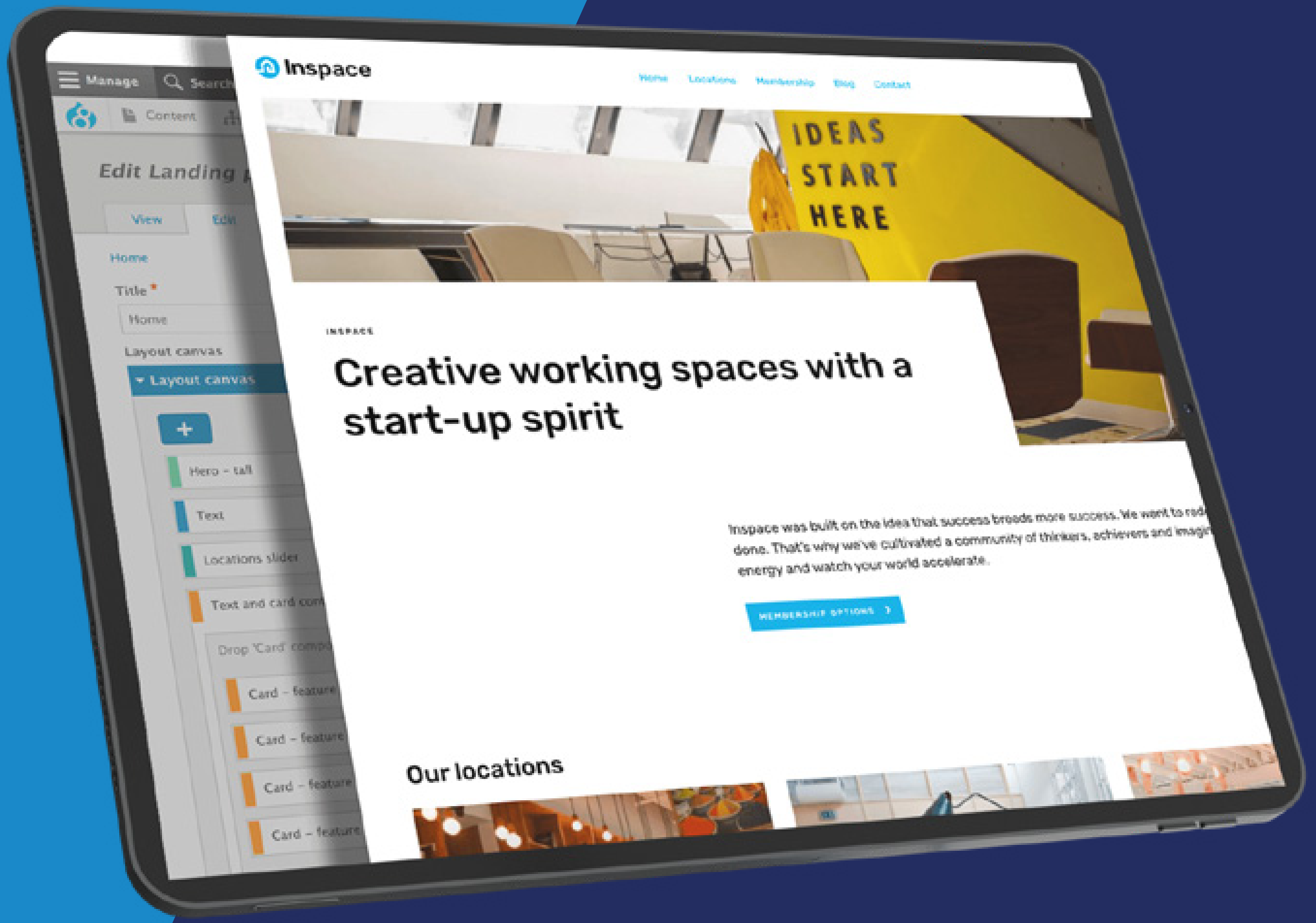
When you export a component, it will take any styles, fonts, colors, and other assets it has a dependency on and create a package file that can be imported into another website. Using this technology, you can create a component-based Drupal website that acts as a living design system. From this, you can export components and import them into other projects as needed.

Conclusion

When it comes to building websites or other digital projects, component-based design systems give designers, developers, product owners, and project managers an effective and efficient way to develop and manage design – especially at scale.

Using components in Acquia Site Studio can streamline the process of building websites and web pages. To see what efficiencies they can lend to your creative, production, and development processes get a Site Studio demo to try it out for yourself.

Request a demo



Creative working spaces with a start-up spirit

Inspace was built on the idea that success breeds more success. We want to red
done. That's why we've cultivated a community of thinkers, achievers and imagi
energy and watch your world accelerate.

ACQUIA.COM

About Acquia

Acquia empowers ambitious digital innovators to craft productive, frictionless digital experiences that make a difference to their customers, employees, and communities. We provide the world's leading open digital experience platform (DXP), built on open source Drupal, as part of our commitment to shaping a digital future that is safe, accessible, and available to all. With Acquia DXP, you can unlock the potential of your customer data and content, accelerating time to market and increasing engagement, conversion, and revenue.