

How to prepare content for

# Sitecore Projects with GatherContent



---

# Table of Contents

**4 Introduction**

Guide overview

**5 Step 1**

Define your content structure

**9 Step 2**

Building out your templates

**14 Step 3**

Building your sitemap

**18 Step 4**

Defining your workflow

**21 Step 5**

Kicking off your project

**23 Step 6**

Tracking your project

**25 Step 7**

Migrating your content to Sitecore

**27 Step 8**

Content Governance

# Guide Overview

GatherContent is the perfect pre-CMS partner, you can use it to produce, store and manage your content before your website is ready to launch. As soon as the basic content requirements and structure for your website are defined, you can start producing content in GatherContent and send it to stakeholders for review and approval.

In this implementation guide, Jennifer Davies from one of the world's largest interactive agencies, walks us through the 8 steps Razorfish follow to using GatherContent for Sitecore projects.

If you are involved in planning, producing or importing content to Sitecore websites, then this guide (and GatherContent) is for you!

---

**Step 1**

# **Define your content structure**

One decision you'll need to make prior to producing content in GatherContent is whether you want to structure your GatherContent instance to store content at page or individual component level. The closer you can get GatherContent to match your Sitecore structure, the easier content migration will be.

## Page level vs Individual component level content structure

**Page level content structure** is when content for a web page is created from one Item in the CMS/GatherContent.

**Items** are the core of your projects in GatherContent. Items reflect your content sections and are where content is outlined, gathered, discussed and edited. When you first create a project, you'll be asked to create some Items.

**Individual component level content structure** is when content for a web page is built from snippets of content stored in multiple component or databases in your CMS and multiple Items in GatherContent.

Let's get into a little more detail about each of the options so you can make an informed decision:

### Option 1: Page level Content Structure

If you are working on a small, highly structured website where content is stored at the page level and content reuse is minimal, you will likely want to structure your GatherContent implementation at the page level. That way, your GatherContent

instance maps directly to your sitemap and it's very easy for your content creators to understand where their content will appear at any given time.

**Pros:**

- › It's easier for content writers and reviewers to visualise the page and how the content will look in context
- › If your website is page-based with limited personalisation, it's much easier to set up and maintain your GatherContent implementation

**Cons:**

- › This option doesn't support content reuse – content entered on one page will need to be copied over if you want to use it again on another page
- › Pages can get quite long to scroll, especially if you add content creator guidance for each piece of content
- › Component settings can be tricky to copy from page to page (although there are strategies to avoid this)

## **Option 2: Component level Content Structure**

If you have several versions of content that could appear on a page, and multiple components with specific structure to enforce as well as tagging to track, I recommend structuring your content at the component level. This will give you significantly increased flexibility should designs shift. It will also encourage your content creators to think more flexibly about where their content might appear and avoid saying things like “see below” as they might not be determining what appears below.

**Pros:**

- › It's easier to maintain component-level requirements in templates, and make changes to just those components as needed
- › It supports content reuse
- › It's flexible to page layout changes
- › It generally loads faster in GatherContent
- › Deeper content guidance can be provided by component without creating an endless page of fields

**Cons:**

- › It requires a significant level of content management and maintenance, and some level of mapping updates if you want to keep components organised by display page
- › Content creators can get confused as to where the content appears in the user journey

Made your choice? Let's get building!

---

**Step 2**

# **Building out your templates**

Before you can start creating Items and loading content into components or pages, you'll need to create some templates to get started. Here are the things to focus on:

## Structure

- › What is the general structure of your component or page?
- › What text fields are needed?
- › Are there character count limitations?

## Images

- › How many images are needed?
- › Are they optional or required?
- › Is there a particular size ratio they should follow?
- › What formats are allowed?
- › Can they be attached or do content creators need to put images elsewhere?

## Content Links

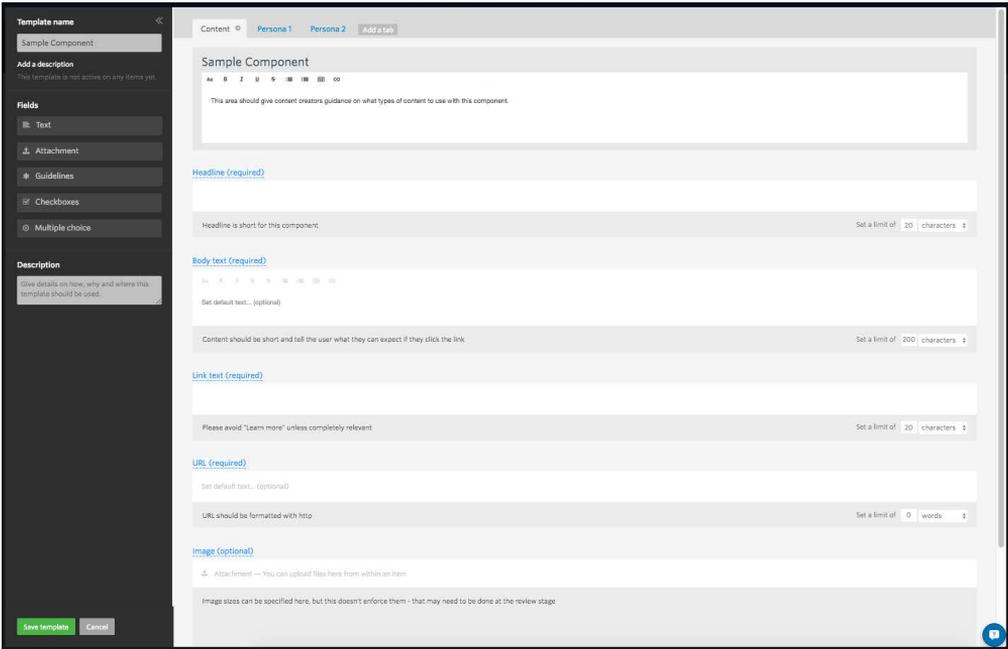
- › Does the content link anywhere?
- › Can you set the link format rather than letting users choose?
- › What are the link text limitations?

## Guidelines

- › Can you add a screenshot of the component or page to the Notes to help content creators visualise what they're building?
- › Can you add content guidelines to the Notes to ensure the copy is in the right tone of voice and style?

The more you can do to get your structure set at the template level, the better chance there is for your content to look great once it's available in Sitecore.

**Here is a sample template:**



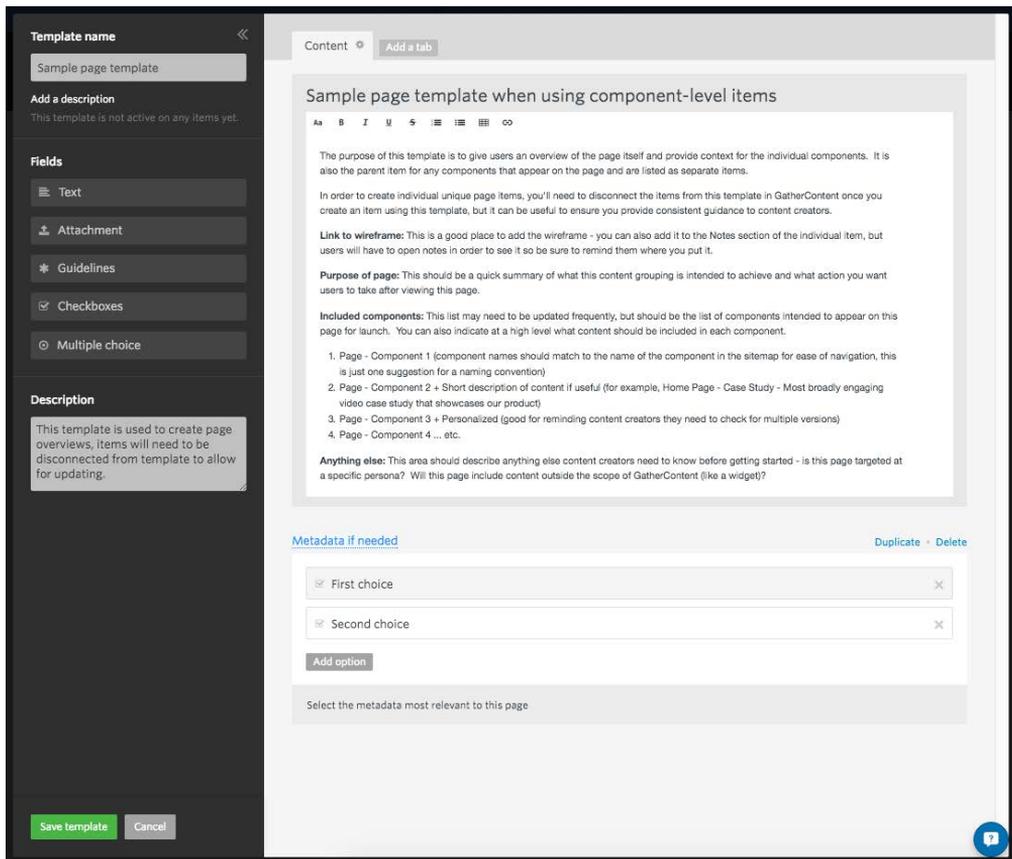
As you can see, there is the option to provide guidance about the template in each field – if you need users to structure their URLs in a specific way, you can let them know that here, or if images need to be a specific size, you can add that to the guidance.

If you have personalised versions of content, you can either add more fields to this view for each personalised version, or create separate tabs for each by duplicating the main component page – that way, all of the guidance and settings are retained, just the content can change:

---

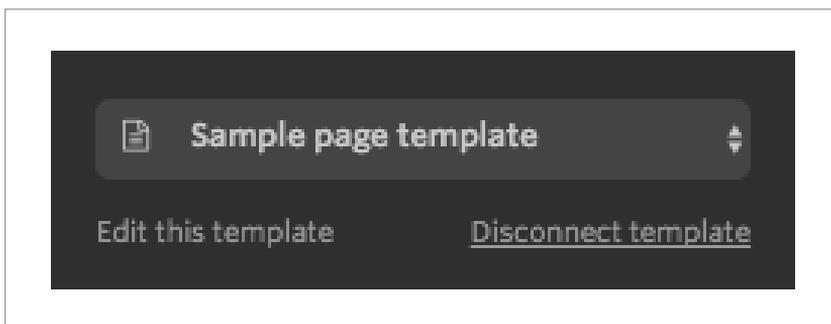


If you are using component structure, you may also want a page-level template that groups together relevant page-level information for content creators including basic page layout where the components might show up as well as provides guidance as to the purpose of a given page.



This sample page template gives an overview of the types of information you might want to include.

Note that if you do have a template for page-level guidance, you'll want to disconnect the template from the master in order to create the custom guidance for each page:



---

**Step 3**

# **Building your sitemap**

The next step is to create all of the instances of content you'll be managing (as Items in GatherContent), whether component, page or some blend of the two.

You can wait to do this until a final sitemap is defined, or you can get started as soon as you know what some of the key pages will be as GatherContent makes it simple to move Items.

### Notes on Naming

If your sitemap is finalised and you're structuring your Items at the page level, simply name the Items the same as they are called on the sitemap.

If you have some larger components that will have content separate from the main page, those can simply be called <page name-component name/purpose> or some iteration thereof and stored as a child of the parent page.

### Create items

Home page -  
Testimonial  
Home page - Blog post

You can nest items using hyphens to represent your sitemap, e.g.

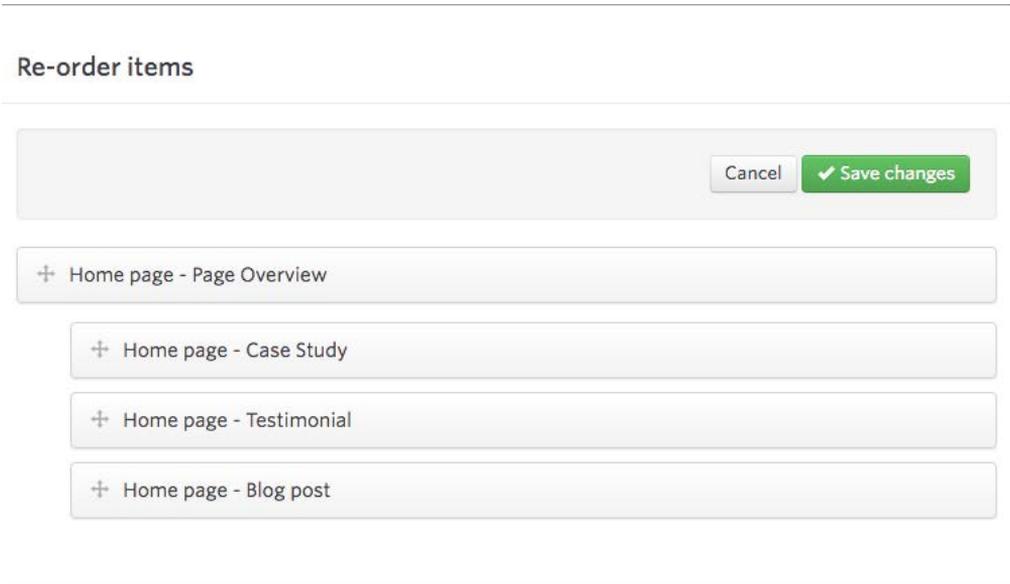
Home  
About Us  
- The Team  
-- John  
-- Mike  
Services

[Add a single item instead](#)

Do these items have a parent? (optional)

Home page - Page Overview  
↳ Home page - Case Study

While adding the page name to the component name may seem extraneous, if you ever want to move the component and you have multiple instances of that component across pages, it ensures you can tell which component goes with which page.



Create Items and associate templates for all of the pages/ components you currently anticipate – this gives content creators and reviewers an idea of the overall site structure and flow so they aren't working out of context.

## Home page - Case Study

● Draft ▾

🕒 Set a due date



☁ Save

▾ Home page - Page Overview

Home page - Case Study

Home page - Testimonial

Home page - Blog post

If you have only components, name them with the area or content type they are grouped by as well as type.

---

---

**Step 4**

# **Defining your workflow**

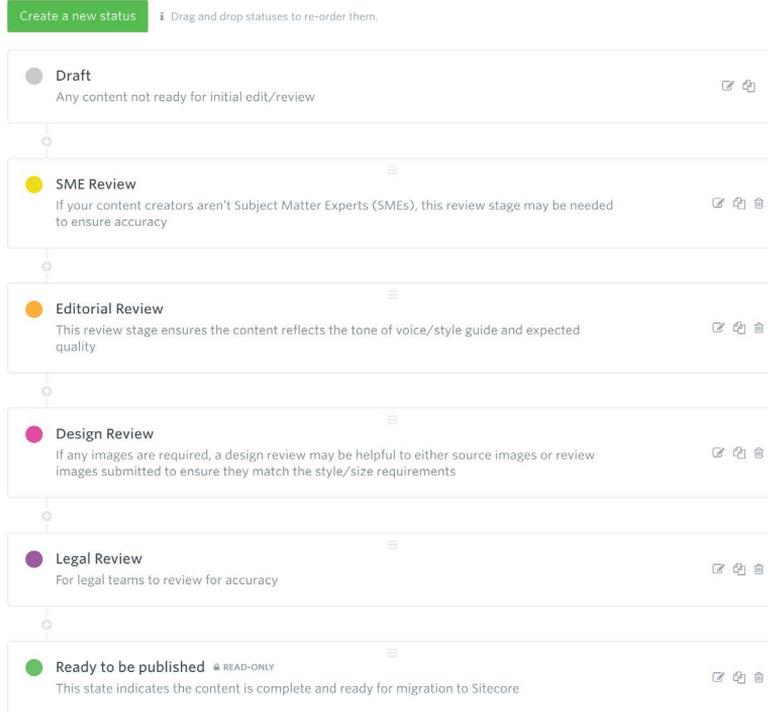
Your workflow should reflect the actual stages required within your organisation to sign off content. All of those users can get access to GatherContent to sign off. You can also control what content and features users get access to so that people only see what they need to.

Your workflow will likely follow something similar to this:

1. **Draft** – Any content not ready for initial edit/review
2. **Subject Matter Expert (SME) Review** – If your content creators aren't SMEs, this review stage may be needed to ensure accuracy
3. **Editorial Review** – This review stage ensures the content reflects the tone of voice/style guide and expected quality
4. **Design Review** – If any images are required, a design review may be helpful to either source images or review images submitted to ensure they match the style/size requirements
5. **Departmental Review** – You may need a review stage for specific departments like Legal or HR
6. **Ready to be published** – This state indicates the content is complete and ready for migration to Sitecore

## Workflow

This is your project's workflow. Statuses can be applied to any item in this project. You can create new statuses and customise the existing ones.



Walk through your workflow with all reviewers to ensure they know what is expected and when. Do you want them to edit the content if they find an error? If not, should they assign it back to the original writer or someone else? Do you want them to change the workflow state if they approve the content? Answering these questions in advance ensures you won't have a team of confused people sitting on content in various stages of completeness delaying your project launch.

With your workflow complete and signed off, it's time to kickoff!

---

**Step 5**

# **Kicking off your project**

While GatherContent is really easy to use if you've structured your templates and Items well, allowing content creators and reviewers to get to work right away, you should still have a proper project kickoff.

Your agenda might include a review of the following:

- › Goals of the website
- › Basic sitemap
- › Structure you've created in GatherContent and how that maps to the sitemap
- › Quick overview of how to use the tool including collaboration features such as commenting
- › Workflow stages and what they mean
- › Expectations from each team to complete the project

---

**Step 6**

# Tracking your project

As long as your team members are using the GatherContent workflow stages correctly, you can track progress in seconds from your project home page.

---

### Project status



- Draft (1)
  - SME Review (2)
  - Editorial Review (2)
  - Design Review (2)
  - Legal Review (1)
  - Ready to be published (9)
- 

If you start to see anything strange, like large chunks of content stuck at Legal Review, or never moving from Draft, check with the teams to see what's up. They may still be confused about when to update the workflow. Remember, this is a one-time project, and probably not their day job, you may need to nudge things along periodically to ensure everyone is meeting their deadlines.

---

**Step 7**

# **Migrating your content to Sitecore**

You can either wait until all of the content is signed off and complete, and do one big migration, or migrate as you go. Work with your development team to understand which one works best for your project.

For small projects you can simply copy and paste into Sitecore but for larger projects GatherContent's Sitecore connector is ideal. Using the connector you can specify mappings and define which templates and fields to import to Sitecore. You can import 100s of Items in seconds - a big time saver! Plus you'll minimise human error. The connector also supports backward connection, so you could update the status of all Items being imported.

You can find all our Sitecore Integration documentation here:  
<https://gathercontent.com/integrations/sitecore>

---

**Step 8**

# **Content Governance**

Use GatherContent to govern your website's content on an ongoing basis. GatherContent's Sitecore Integration offers an update feature, which is fully compatible with the Sitecore workflow. You can also update GatherContent's workflow once your content is live, setting deadlines for reminders to update content and better suit ongoing content collaboration.

*“Using GatherContent to produce and approve web content ensured that content was ready to go as soon as Sitecore was - in the right format, size and shape. We saved a lot of time!”*

**JENNIFER DAVIES**

*Razorfish, Content Strategist*



# Painlessly plan, produce and publish website content

Content collaboration designed for web, marketing and communications teams.

Learn More

No credit card required

