How to build, manage, and get ROI from your SaaS knowledge Base

A ConversionTown guide

How to build, manage, and get ROI from your SaaS Knowledge Base

This ebook is for any SaaS company who knows that calling it a “Knowledge base” doesn’t really scratch the surface of what’s possible.

Along with knowledge, you get improved customer experience, taking people from “Argh how do I do this” to “Ahhh that’s better”. There are marketing opportunities too, from showcasing epic features and how simple they are to use. Plus plenty of productivity boosts, as you save hours on answering user questions through articles and maybe AI chatbots.

That’s all covered here. You’ll find explanations, how-to’s, and tips and things to avoid. All based on real-life knowledge base builds, article creation, and AI setups.

Enjoy the read,

Steven Kent,

Content strategist, technical writer, optimiser, Mayor of ConversionTown

steven@conversiontown.com

020 8088 8797 / 07396 576 337

conversiontown.com

Part One: Planning your knowledge base

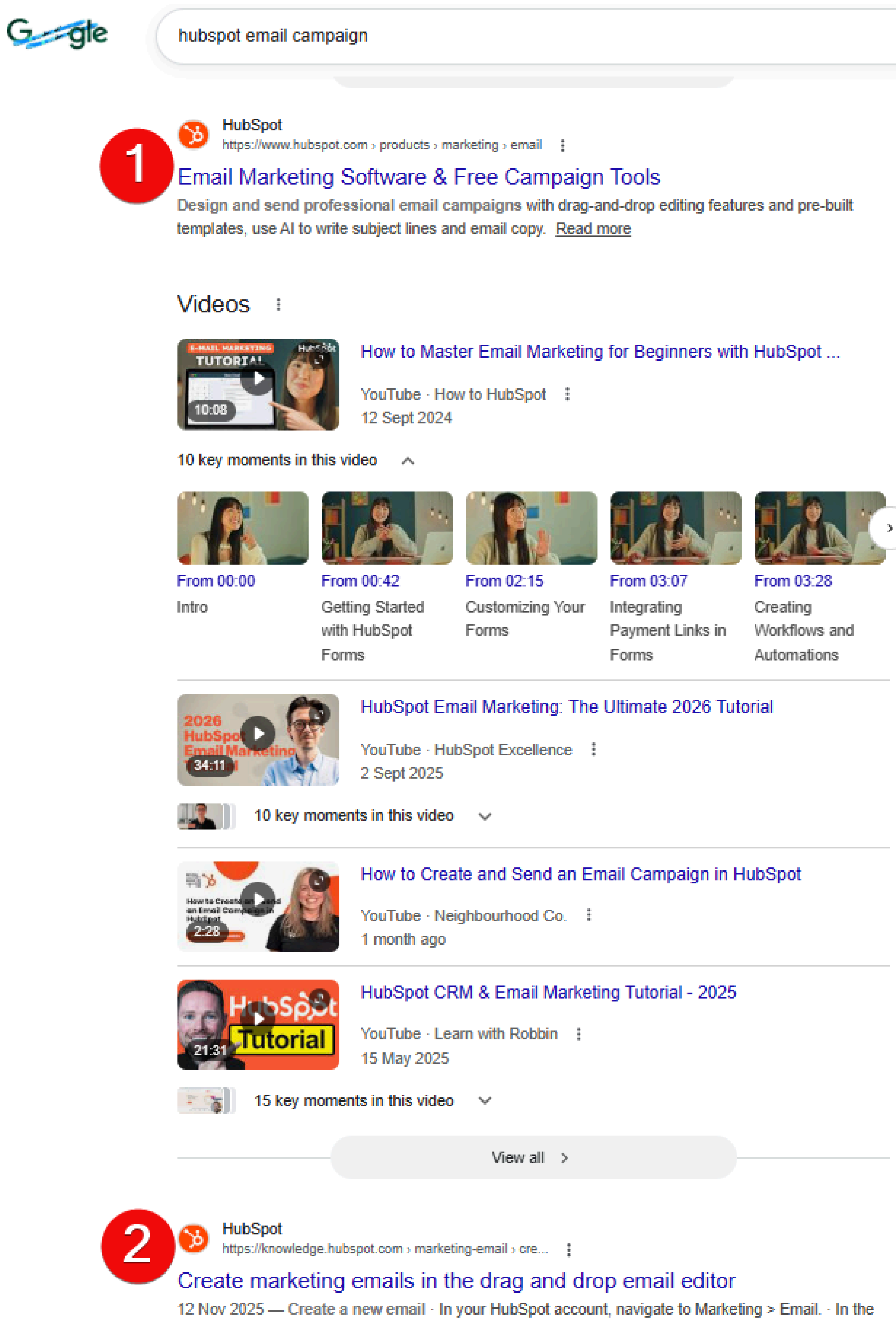
Strategy about structure

First, work out what you want your knowledge base to do. Maybe one of or all of these:

- **Letting users know about a new feature?**
  If you've got segmented data on users, it might be time for a newsletter. People read what's interesting to them. So as long as you're telling them about new features and not starting with "we're delighted to announce…"
- **Onboarding with a walkthrough?**
  Something with visual shots of popular features for general walkthroughs, or announcing new features
- **Reduce tickets being raised?**
  Some sort of chatbot can help to act as a first line responder.
- **Increasing traffic from search?**
  Just be sure which page you'd like to appear higher. A sales page about a feature, or a help article about the same feature?

Then pick some metrics to measure the success of your strategy.

If you want to...	**Metrics to use**
Promote new features	• How many users sign up to the feature • How many users use the feature
Onboard users	• Page views of the release note • Clickthroughs from the release note
Reduce tickets being raised	• Number of tickets raised before and after a feature is introduced/improved • Customer feedback
Increase traffic from search	• Page views filtered by source • Bounce rate



HubSpot have the marketing page (1) above the knowledge base article (2), with videos sandwiched in between to cover all angles

Filtering your audience

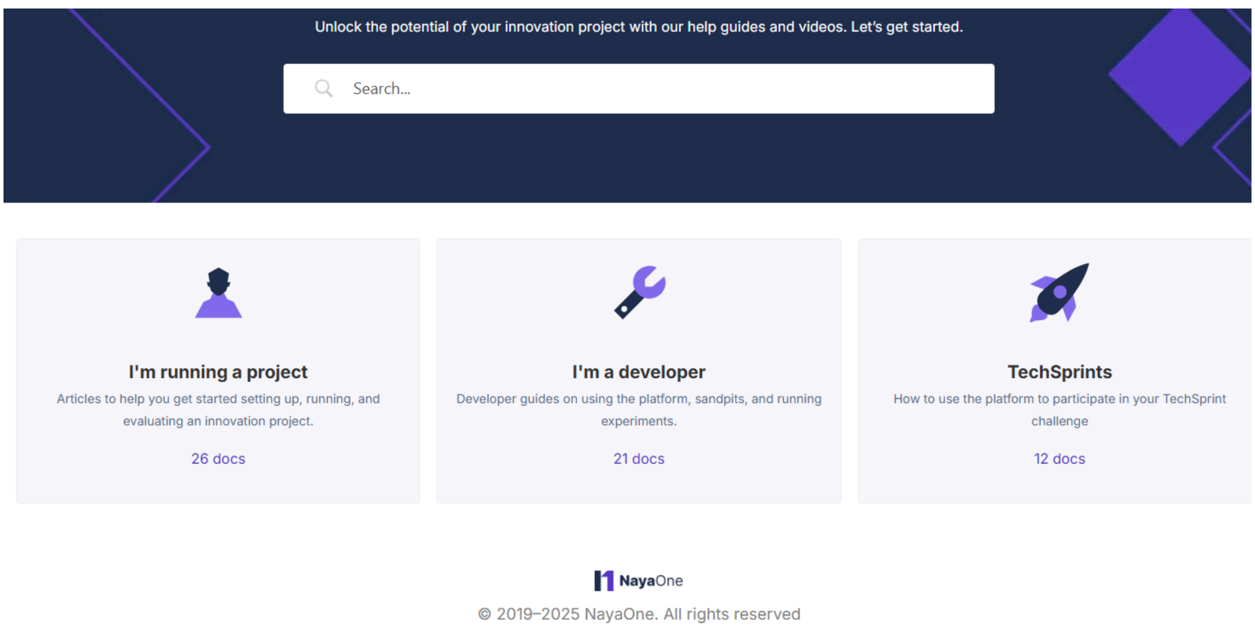
If you’ve got multiple audiences, such as developers vs marketers vs admins, you’ll want to funnel them when they land on your knowledge base home page.

If you’ve got one main audience, you can filter by knowledge levels. Getting started vs advanced features.

It can be as simple as having buttons for each function or skill:

I’m a developer vs I’m a marketer

Getting started vs Advanced email workflows



Here’s how I segmented the audience when creating NayaOne’s knowledge base

---

Information Architecture (How to organise your knowledge base)

There are three main approaches to structuring a knowledge base. Sometimes with a bit of crossover between each one, which is when categories and tags can be useful. After all, it's better to have one article get found in two places rather than not found at all.

User levels

Based on their expertise and experience levels. Usually works best for products with limited features, eg Getting started vs more advanced use cases.

Product features

If you've got a payment system, you might want sections like "integrations with payment gateways" and "registering users".

Role related

This can overlap between the role (I'm an admin vs I'm an editor) and function (I want to add a user vs I want to publish content)

Just don't create categories as you go; decide on a few before you launch. Or even better, test them out with some customers to make sure you're using the same words that they do.

Content Planning (and what to prioritise)

There are two approaches, depending on if you've already got a knowledge base...

Approach 1: With an existing knowledge base

Auditing Existing Support Tickets

1. Put them all into a spreadsheet and upload to an AI (taking care to remove any sensitive data or GDPR-esque risks)
2. Segment (yes that word again) your support requests into themes.
3. Look for any patterns, such as questions asked or complaints. These patterns tell you where to start. Sometimes you find quick wins from just fixing a few articles with more copy or updated images. Other times, it's a case of 'ah we need to create articles on this'

If you've got analytics, organize into what gets the most views or interactions, and work your way through those first. Then mark each article with either:

- Keep
- Update
- Delete

Go through any support tickets and customer enquiries. Any issues coming up that don't have an obvious help article? Add those to your audit spreadsheet, with the category “needed”.

Approach 2: Without an existing knowledge base

Not yet got a knowledge base? Time to rely on your customer insight and make a list of articles needed. Then spy on your competitors’ knowledge bases. Compare your list to theirs and see what’s missing.

While you’re there, see what their articles are missing and make sure yours are better. Maybe better images/gifs, a different tone of voice, or simpler taxonomy?

Article types

- **Getting started**
  These are about giving users an ‘Aha!’ moment as quickly as possible. So pick some key tasks, ideally fairly easy (hyperlink to more in-depth articles for users who want to go deep early)
- **How-To**
  The most common type of articles**.**
- **Troubleshooting**
  Include reasons why something’s happened, and be very specific about how to fix things
- **FAQs**
  Be careful with these. FAQs are often the de facto response to “I don’t really know where to put this article”, and can make your knowledge base bloated and hard to use. Often they work better within articles.

For example: “Upload a profile image (maximum 10GB and 300 px X 300 px), instead of having a separate FAQ article asking “What’s the maximum file size for uploaded images”

What doesn't belong

This might need a chat with your marketing team and/or devs, to keep them on your side.

It also needs you to be able to ‘own’ the help centre, even if you’re getting pressure from other teams or higher-ups.

But if you want to keep your knowledge base lean (which you do, for users and for any AI integrations), then set out some rules:

- **New product announcements:** Put on your website, with a link to a how-to article if the feature is live
- **Release notes:** Put on your website or create a section purely for release notes and link to them in your knowledge base footer and in emails to users
- **API documentation:** If more than 3 articles, create a microsite with its own articles and tone of voice

Part two: Writing help articles that help

Titles

Start titles with either “how to” (how to upload data) or action verbs (“uploading data”).

“Understanding” can be useful when you’re issuing some type of information, such as “understanding your monthly usage”.

Do some analysis of the language your customers use, and mirror that. “How to change my profile picture” is better than “How to modify my profile picture”.

Intros

Add 1 or 2 sentences summarising the article. Include any prerequisites, such as “pro licence” or “administrators” either as a label that appears above the article (Zendesk offers this), or use a pullout box in a different colour.

For long articles, add anchor links for users to expect some lengthy detail and also allow them to jump to what they want to know.

Steps

Unless the user journey is very obvious/simple, keep to 3 or 4 actions per step. So something like:

1. Click Settings > My profile > Profile picture

   (depending on your UI’s complexity level, here you could add a screenshot)

2. Click Update picture

Include some kind of finishing step so the user knows they’ve arrived at the end. For example:

You’ve now updated your profile picture

Words/phrases to avoid

- **Navigate to...**
  (unless you’re a knowledge base for ship crews, use “go to”)
- **Complete the following steps**
  (the user can see the steps and guess they should follow them)
- **“Click on”**
  Just use “Click”

A word or 3 on screenshots and gifs

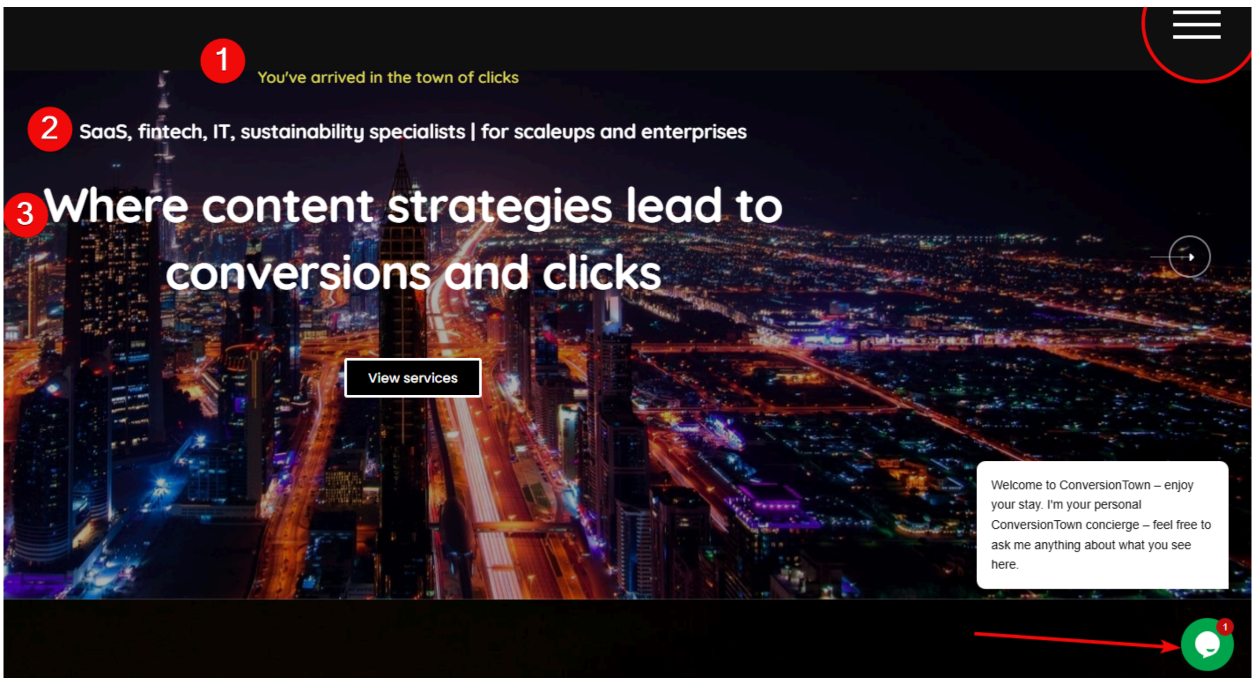
Use screenshots when the UI is complex or unfamiliar, when the user, or when a visual reference would save significant explanation. Skip them for simple actions like 'Click Save' or when the UI is self-explanatory.

And use circles or arrows (decide when to use each, eg arrows for where to click, circles for highlighting areas) on screenshots.

Use gifs for short steps, such as opening a dropdown menu or clicking hard-to-find icons.

Create a system for saving filenames, so you can find and reuse them instead of wasting time and storage on recreating for each article. For example, category-featurename or section-featurename.

Videos are a bigger investment, so keep these for features that drive the most ROI and that aren't likely to change that often. The last thing you want is showing users out-of-date videos.



For screenshots and gifs check out ShareX or Snipping Tool. Free, and with features such as arrows, magnify, steps, blur, pixelate, record, and lots more

---

Tone of voice

Always imagine your reader is frustrated. They're not on your knowledge base for pleasure; they want to solve a problem or find a solution. That means:

- **Plain English**
  Want everyone to understand you? Write for primary school ages
- **Straightforward words**
  Write “use” or “help” instead of “utilize” or “assist”
- **Supportive language**
  Use “avoid” rather than “do not” where possible, eg “Avoid adding too many fields” rather than “Do not add too many fields”
- **Confident tone**
  Avoid using “if you would like to” and “do not hesitate to”. Go for “To try this feature, contact your account manager”
- **Short sentences**
  Keep to below 20 words, readers find it hard to keep track especially online
- **Personal, 1:1**
  Write as if you’re talking to one person. “You can also set up user accounts” instead of “It is also possible to set up user accounts”

GOV.UK should be an authoritative, trusted source. This means we need to write in a way everybody understands. We know people distrust jargon and that being clear and direct helps - as do shorter sentences.

Writing guru Ann Wylie describes research showing that when average sentence length is 14 words, readers understand more than 90% of what they're reading. At 43 words, comprehension drops to less than 10%.

Studies also show that sentences of 11 words are considered easy to read, while those of 21 words are fairly difficult. At 25 words, sentences become difficult, and 29 words or longer, very difficult.

Source:
https://insidegovuk.blog.gov.uk/2014/08/04/sentence-length-why-25-words-is-our-limit/

Making content findable

However your users like to search – AI agent, basic chatbot, or search bar – it's still about getting the basics right. That means making sure your knowledge base works both for humans and AI. And that means:

- **Headings**
  Break up articles with headings and subheadings
- **Intro text**
  Write for people scanning, in a hurry, maybe stressed or annoyed. That means putting plenty of relevant keywords in your title and first paragraph.
- **Tags and categories**
  Let users explore and view articles by topics if they want to
- **Hyperlinking**
  Link the first mention of the word or phrase, to let users explore and AIs (if you're using them, which you

should) to crawl and understand articles that are related

- **Terminology**
  Look at how your users describe your platform. Make sure you use the words they use, like 'delete my account' rather than 'Account closure process.'

Of course, now your customers are used to AI, you need more than that.

Got a basic search? Upgrade it to something like Algolia (I'm not an affiliate, I just use them a lot and they've never let me down.

Search "Ai search bar" for alternatives). It's a search function that's superfast, takes care of semantics, and plugs in without any fuss.

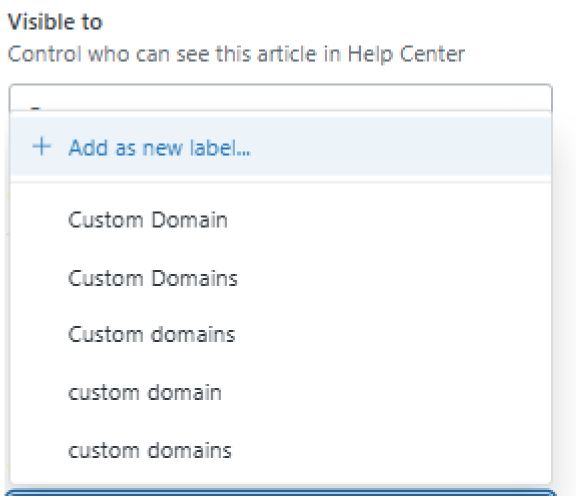
Or go one step further and [integrate an AI agent/chatbot] (here comes the pitch: That's something I've done with Intercom and Fin, and can build using tools like Voiceflow).

Tagging and Metadata

Be really really really strict over who can create these. Otherwise you'll end up with multiple tags like Artificial Intelligence, artificial intelligence, AI, ai.

Check where meta descriptions will be shown and make sure you meet the character limit. No user wants to see "In this article we'll show you how to…" and then have to click through to see what it's about.

Oh, and agree on the spelling and case to keep things consistent.



Limit who can create categories, and make sure all contributors have a style guide to refer to

---

Part Three: Managing and maintaining your knowledge base

Ok, you've got the content finished and updated. Your knowledge base has launched.

Everything is up to date.

And then... new features get introduced. Fixes and improvements get made to the UI. Menus get rearranged, new buttons get added, and soon your knowledge base is out of date.

So who's going to update it? There are tradeoffs with each option:

- **Customer Support?**
  They know exactly where customers get stuck and how they phrase their problems. But sometimes that makes it harder to explain things to people without much knowledge.
- **Product/devs?**
  They may write content that's accurate, but it doesn't come from a UX perspective or speak directly to pain points
- **Technical writers?**
  They may not have the day-to-day relationships with support and product to know what's been changed and what customers are saying.

🚨Pitch alert: I'll update it for you; just message me at https://conversiontown.com/contact

Publishing and proofing

You'll need some sort of reviewing process to ensure accuracy and consistency. But you also don't want to create bottlenecks. Especially when your dev team has worked all weekend on a release and they want users to know how to use it.

Plus you want to empower those who do have writing responsibilities. So try a tier system:

1. **Sensitive content**
   Needs a review before publishing, maybe from compliance, legal, or PR

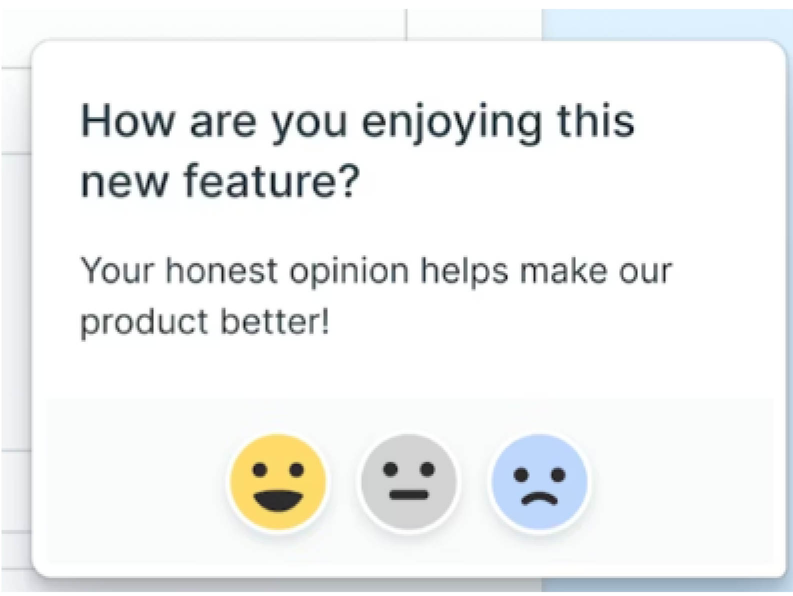
2. **New articles on features**
   Need a quick accuracy check from someone on the team that built it.

3. **Updating images, fixing typos**
   Publish without extra checks

Optimising

Make sure any enquiries or customer questions are seen by whoever maintains the knowledge base. So they can know of any missing content, outdated pictures, or confusing messaging.

Many systems invite feedback at the end of the article. Thumbs-up or thumbs-down can help, but a comments box is where the treasure is usually found.



These sorts of emoticons don't give you useful feedback on their own. I've had articles on email deliverability get downvoted by users, and it's turned out that it's because their emails are spammy, rather than that there's anything wrong with the article. Add a comment box to get more context (and know if a user is just having a bad day)

Updating

Ideally, you'd update articles whenever there's been a change.

In the real-world, you'll need to schedule audits. The frequency will depend on how many articles you have, resources available for editing, and how often new products and features are shipping. But try to do it every 3 months at a minimum. Check by:

- Doing walkthroughs of how-to articles, checking if steps are still correct
- Running a check for broken links
- Reviewing images
- Checking if the feature mentioned still exists

Removing articles

Create a process for what happens when a feature is removed or is being sunsetted. Your users will probably still look for it, either in the UI or your knowledge base, so update the article saying:

- This feature is no longer available
- Here's what you can do instead (link to the replacement feature or an alternative)

If it's visible and indexed by search engines, apply a redirect for at least 12 months. Monitor your analytics and when you're sure users aren't looking for the article anymore, archive it.

Managing your media library

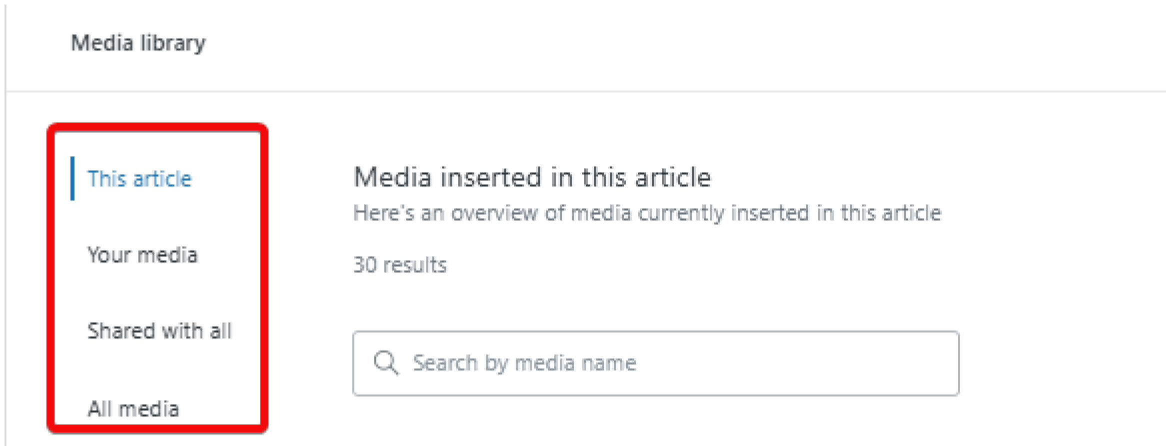
The biggest headache with images is when you need to update all images after a brand change or to reflect a new UI.

Depending on your platform and tier, you can update an image centrally, and that updates the articles where the image appears, which is great. Zendesk Support Enterprise Plan does this.

Document360 has something similar, where you work with reusable content blocks. Otherwise, you might have to do it one by one, which is not great.

That's why it's important to develop a naming system for your images, so you can easily find which articles need updating. Try something that includes categorisation. For example, a screenshot of the billing page in the account settings section could be account-settings-billing-page.jpg.

This also helps with storage. You can reuse an image already uploaded, rather than recreating and reuploading. For example, with Zendesk, you'd look for an existing image in **All media** rather than creating and uploading a duplicate:



To save further on storage and re-edits, try to limit the number of images you use. Sometimes one gif showing three clicks can be better than three images showing individual clicks.

Otherwise, managing the media library is an ongoing task. A bit like that person who paints the Hungerford Bridge, who by the time they finish at one end has to start painting it again at the other end.

Measuring what matters

We've covered setting up and maintaining your knowledge bases. Now let's find out how your knowledge base is performing. That means picking some core metrics:

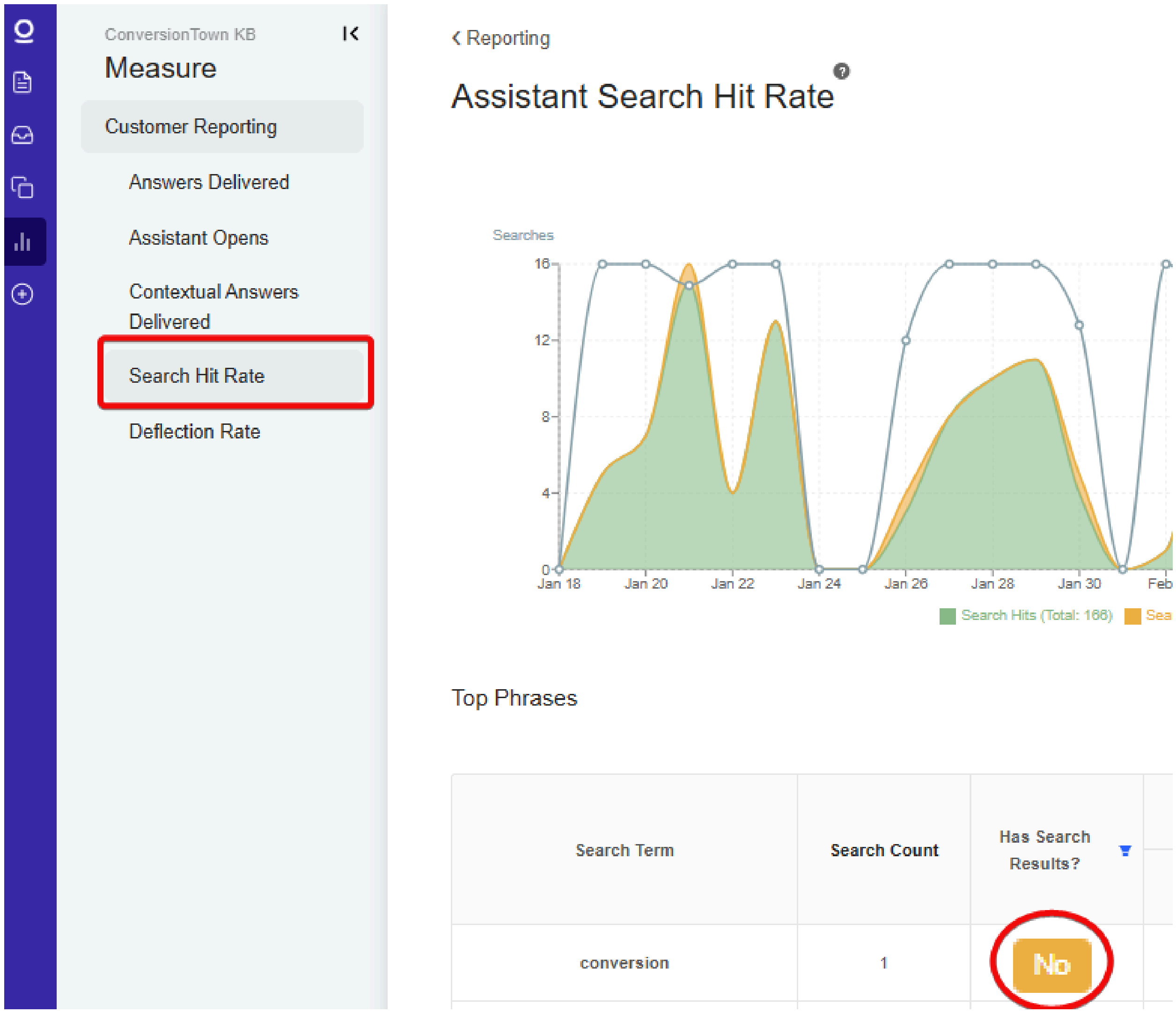
- **Article views**
  High numbers can mean they're popular topics, or also that they appear high in search results.

  Meanwhile, low views might mean the topic isn't popular, or that the article is hard to find. So always confirm your hypothesis before making any strategic decisions on what users are most or least interested in.

- **Search bar**
  This is your goldmine. You can discover the words people are using and searching for. And you can see if any queries are returning 0 results. Fix by verifying you have keywords and tagging set up correctly.

- **Feedback**
  Try to go beyond the standard feedback many platforms offer. The thumbs up or sad face doesn't tell you much on their own. An article on email delivery might be great, but if the user is getting marked as spam, they're more likely to give your article a downvote. So invite comments (with some sort of moderation rather than allowing instant publishing) or ask for detailed feedback.

- **Calls to action**
  Set up tracking about articles that refer to add-ons or any service that needs an upgrade or subscription. If your knowledge base is bringing in business (and it should), you need to know.



Screenshot from Elevio, showing terms that users searched for but didn't get any results for

---

Part Four: Choosing your platform

Ok, you know what to do for your technical content. Now you just need to decide which platform will do it best. For the content creation part, there's not much difference in features. All the ones I've used (and cover below) have all the writing and editing tools you'd expect to find.

A bigger influence on any decision (apart from budget, more on that below) is the technical side of things.

Especially integrations.

Will you want integration with your help desk ticketing system? How about your CRM, to understand more about individual customers? Perhaps some advanced analytics to understand your ROI in greater detail?

Of course, custom-built gives you maximum flexibility but also the maximum amount of work too.

If you want a quick answer/recommendation, an off-the-shelf solution with some integration gives you the best of both worlds.

Pricing and (sometimes hidden) costs

To work out a budget for the platform, decide on:

- How many people will upload content? (and how many licences will you need to pay for)
- Do we have in-house expertise for setup or do we need to get external support?

- Do we have to pay for a knowledge base custom domain?
- Do we have to pay every time a user uses the search?
- Do we have to pay fees based on article page views?

Zendesk

Good if you're on Zendesk Suite because you can connect your knowledge base talking to your support tickets. You can also plug into external forums for community/crowdsourcing knowledge.

Also comes with Answer Bot, the AI that can take questions from customers and suggest articles based on what they're asking.

Now the downside; it's expensive. You're paying for the whole Zendesk ecosystem even if you only want the knowledge base bit.

Customising the appearance will probably involve custom CSS for anything beyond the basics. So it might eat up some of your developer resources.

Intercom

Fin has been a game-changer in terms of Intercom. The AI inside Intercom can suggest articles when your users are chatting with Fin. No need to exit the chat or switch screens.

There’s more of that fluidity because your articles can appear inside the Messenger widget, get attached to product tours and tooltips.

If you're already using Intercom for messaging and support, articles can be connected pretty easily. But if you’re not already using Intercom, it might be a bit expensive.

Help Scout Docs

Help Scout has far fewer bells and whistles, in a good way if you’ve got a fairly simple SaaS. Otherwise for scale-ups and enterprise, I’d advise looking elsewhere.

Taxonomy options are limited, but you’ll be fine as long as you make the right decisions at the start around information architecture.

It also offers a widget (named Beacon) that lets you embed help into your product with article suggestions and a contact form.

BetterDocs (WordPress)

This is a WordPress plugin that can either turn your site into a knowledge base, or allow you to add on a knowledge base.

If you already use WordPress, there’s less of a learning curve. However, the default branding options are limited, and you'll need some dev support if you want to go beyond the basics.

You also need to take care of updates, as you do with a normal WordPress installation.

Quick comparison

Feature	Zendesk	Intercom	Help Scout	BetterDocs
Standalone viability	Medium	Low	High	High
Help desk integration	Excellent	Excellent	Excellent	Manual
In-app widget	Yes	Excellent	Yes (Beacon)	DIY
Design flexibility	Medium	Low	Low	Excellent
AI features	Strong	Strong	Basic	None
SEO capabilities	Good	Basic	Good	Excellent
Multilingual	Built-in	Limited	Workarounds	Plugins
Cost	£££	£££	££	£
Best team size	Mid-Enterprise	SMB-Mid	SMB	SMB

Part 5: Post-launch: Keeping the momentum going

You've done all that work, launching (or relaunching), writing and updating. Now it's about keeping your knowledge base up-to-date.

For your users, and also for whatever comes next with AI or another yet-to-be-imagined technology.

That’s when you need to:

Get Buy-In

Promote the knowledge base across departments. Show how each team benefits from it being up-to-date. Build relationships with the product team and devs, let them know you want to showcase their work.

Show impact

Share metrics on the knowledge base’s impact. How has it affected the number, or type, of service tickets? Tell authors and others (and their managers) how their contributions have helped.

Review regularly

Reviewing and updating is always better when it’s done little and often. Set aside some time each week to check in with your team, know what’s being shipped and what changes will be needed. Ideally you want all articles to say ‘last updated’ within the past month, but realistically keep this to 3–6 months at the most.

We’re up to page 29 on this. So it’s fair to say that all of the above is a lot of work.

There is another way to do all this...

Yes, time for another pitch. Get someone (me) to take care of it all for you.

That includes any and/or all of this:

- Building the knowledge base
- Auditing what you've got
- Identifying what you need
- Creating the articles
- Planning the layout and structure (categories and tags)
- Integrating an AI chatbot
- Analysing and optimising

"Steve was fantastic to work with - he was organised, thorough, and was really able to take a loose initiative and run with it with himself, seeking out the relevant people in the business and working with them very well. He works to a very high standard and I would definitely work with him again. Huge thanks for your hard work and help Steve!" (view case study)	 **Louis Marcantonio** Product Leader, NayaOne

Let's make your knowledge base do epic things...

steven@conversiontown.com
020 8088 8797 / 07396 576 337
conversiontown.com