Lullabot®

by Matt Robison

Challenges for Large Content Teams

And how to overcome them

Challenges for Large Content Teams: And How to Overcome Them

by Matt Robison

Table of Contents

Introduction

As your content team grows, so does the potential for new problems. Or, as some call them, “opportunities to work better together.” With more editors comes more ways to bump elbows, cause confusion, and miscommunicate. Still, if you work through these potential problems, your organization will be better off than before, with an unobstructed content pipeline that flows freely.

And that’s the end goal, right? A steady stream of quality content gets published on your website, helping your organization reach its most important goals. Then, you expand your team to increase the output to reach those goals faster.

But you want to be aware of the potential problems that larger teams can encounter. If you don’t pay attention, growing your content team could have diminishing returns more quickly than you thought possible. Adding more people doesn’t automatically mean “more output.”

What is "large?"

It depends. The threshold will be different based on the complexity of the website, how much work needs to be done, and how your team is organized.

For example, you can have lots of non-technical staff writing content, but if all content is funneled through a single content admin to get it into the CMS, that wouldn't necessarily qualify as large. This scenario may present a bottleneck, but if friction occurs due to your team's size, it won't affect the website since most of your writers don't touch it. Instead, your problems will most likely be process or cultural challenges.

You could also have twenty people, each responsible for a single content type or section of the website. This situation isn't likely to cause many problems because of team size, even if the total number is considered "large." Challenges with permission management and access may arise, but that would depend more on the content model, not the team size.

On the flip side, you could have teams distributed across multiple departments, each responsible for its silo of content. This is a common scenario for many industries, including state governments, higher education, and enterprise organizations. Each team could be large, but the overall number of content editors across the entire organization might not be relevant. Each team could also have a different workload, and some departments may publish more content than others. These setups can create process and cultural challenges, but there might also be technological challenges that prevent your teams from putting out their best work.

Some organizations with teams of content producers and editors working together publish a high content volume, following an editorial calendar about as flexible as Han Solo frozen in carbonite. The teams may also practice some type of "create-approve-publish" workflow. This type of setup presents the greatest risk of becoming out of control because mistakes can cascade, bottlenecks can be created, and the workarounds for those bottlenecks could lead to further frustrations and bumping heads.

And then, there are potential translation requirements that build on top of the primary content flow. Do translators count as part of the content team? Again, it depends. Depending on the workflow, translation and internationalization can add a whole new set of problems, from process problems to cultural problems to technological problems.

These examples just touch the surface of possibilities. For the purpose of this eBook, this definition of "large content team" will be used with the caveat that this might not map perfectly to your organization.

A large content team is five or more people working on a single website whose responsibilities overlap and/or whose responsibilities directly depend upon the actions of the other members of the team.

Two people can usually keep things running without being too deliberate. As soon as you add a third wheel, you start to bump into some of these problems, but people can keep stumbling along and keep things working without intentional, crafted solutions. They still interact daily with each other without having to try very hard. As a result, you'll probably still see a big gain in output. However, once you reach five, the lack of defined processes begins to gum up the machinery of your content pipeline.

The biggest challenges

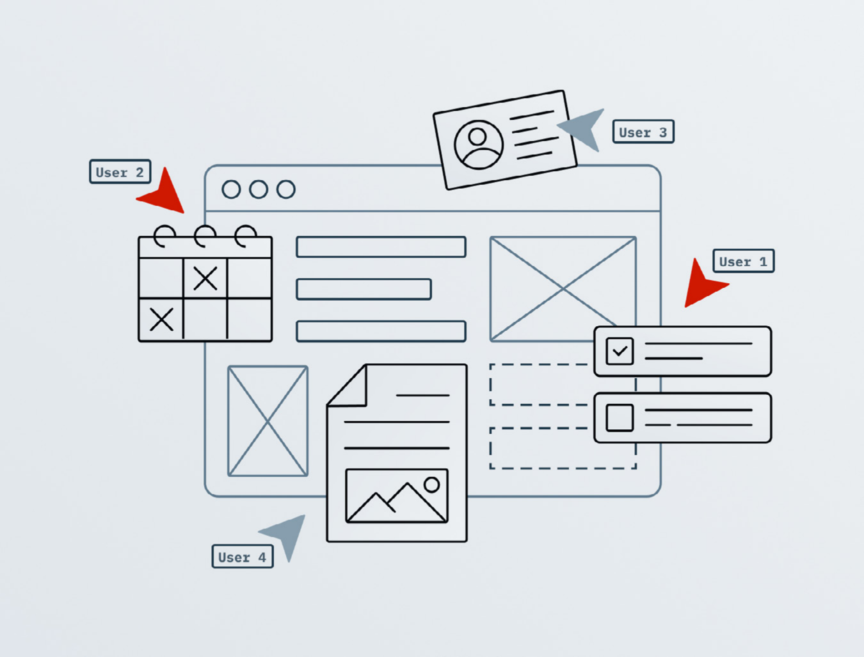
When trying to solve these challenges, it's essential to realize that technology cannot provide a solution to everything. Technological solutions can sometimes help, but not always. You can find some examples below. But often, attempting a technical solution is like pasting over a crack in your foundation with bubblegum.

Many times, a solution requires changes in your process, culture, or both. Unfortunately, these changes are much harder to identify and implement, which is why so many people default to technology.

Let's dive into the problems.

1 Multiple people editing the same piece of content

With idioms like "too many cooks in the kitchen," this problem has clearly reared its ugly head for generations, even before the idea of websites was ever imagined. Now the kitchen is virtual.



Multiple people share responsibility for the same content and edit each other's work. For example, Person A opens an edit form and leaves it open without saving it. Meanwhile, person B opens the edit form, makes some changes, and saves it. Person A returns to their task and saves their changes, overwriting what person B just did. Even if an approval workflow is in place, problems can erupt. Which draft takes precedence?

SOLUTIONS

- **A clearer demarcation of responsibilities.** Should more than one person feel they have the responsibility to change a piece of content? The answer may be "yes," but that means you have some extra work to do in managing potential conflicts. Consider building a content matrix as a first step in the right direction. A content matrix is a shared spreadsheet of pages with URLs, recommendations, page titles, descriptions, and who is responsible for writing or editing the content. You need stronger content governance and more communication. Implementing these can surface additional problems that have nothing to do with technology, which gives you opportunities to improve at a deeper level than you might have planned.
- **Content locks to avoid concurrent editing.** Modules like Content Lock can help, but they should be coupled with a process requirement. For example, implementing notices that display the user who locked the content. The notice informs editors to directly contact the user to unlock the content, which fosters more communication and cooperation.

2 Multiple copies of assets

Editors often re-upload the same image over and over again, even though they could have reused one already present in the system. This leads to resource bloat, and if you have lots of content, it can significantly increase storage costs. And for those editors who do search for assets before thinking of uploading their own, it makes it harder for them to find what they are looking for.

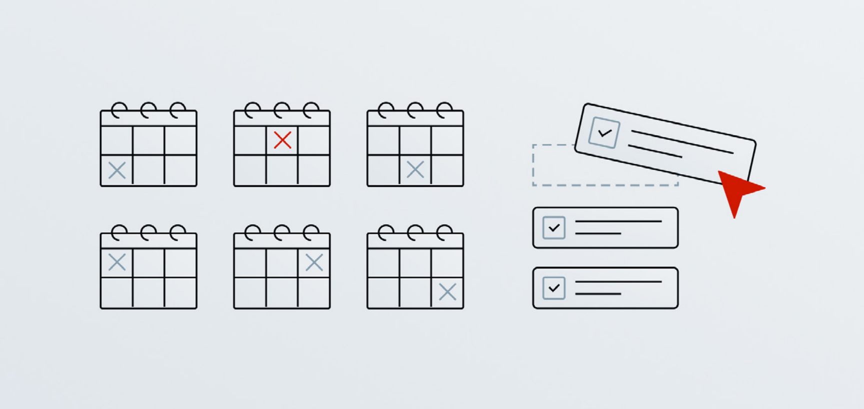
Technology cannot solve the issue of multiple copies of assets. Unless editors change their habits, implementing the most robust, feature-rich DAM (Digital Asset Manager) on the planet won't solve the problem. Assets still need to be tagged and organized so people can actually find them, and doing that requires people, not technology.

SOLUTIONS

- **Training editors.** You can't get around training editors to search for assets before uploading new assets. But this also means that they need training to name and tag assets properly so they can actually be found.
- **Great UX for asset management.** If you are asking for someone to change their habits, the least you could do is make it as easy as possible. A great user experience for searching and tagging assets increases the likelihood of success.

3 Editors who use the website infrequently

Some editors may only hop in every few weeks or every few months. They don't use it enough to grasp the complexities and end up making mistakes, and if they have permissions they don't need, some of those mistakes could be catastrophic. The UX can also become overwhelming.

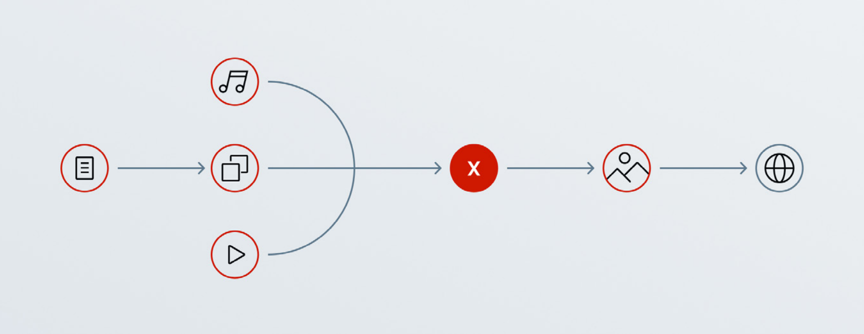


SOLUTIONS

- **Limit the number of editors.** This problem of infrequent users might be a sign that too many people have access to change the website. More rigorous processes might help.
- **Approval workflows.** This is an area where strict approval workflow helps. Nothing gets published unless it has been reviewed and approved by someone who is an expert at working with the website.
- **Easy ways to get help.** Slack channels dedicated to content entry on the website are incredibly useful. Anyone can go there to get support.
- **Simplified UX.** Hide infrequently used form elements under an "advanced" section. Limit permissions to site-wide configuration pages. Create a simple, dedicated menu just for content editors.

4 Workflow bottlenecks

How many gates does a piece of content need to pass through before it gets published? If there are not enough people with approval permissions, things can pile up fast. If one gatekeeper is overtaxed, even smaller teams can struggle to get a small amount of content flowing through the pipeline.

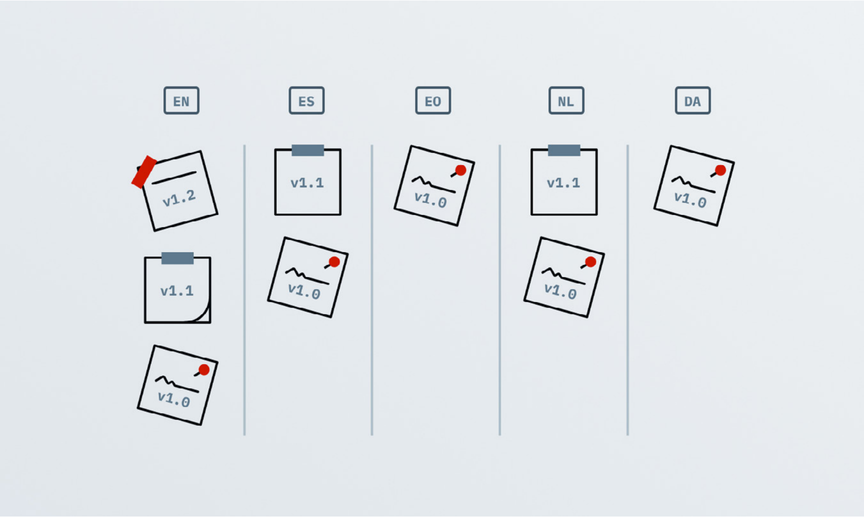


SOLUTIONS

- **Determine if the workflow is really needed.** Is it meeting your needs, or is it just causing more pain? Is it really necessary? Is the workflow there because there is a low level of trust? Is the workflow there because of silly mistakes caused by confusing UX or lack of training? There might be a deeper issue.
- **Separate pipelines and responsibilities.** If everything is currently stuffed into one workflow, see if there is a way to separate them. This works best if content editor responsibilities fall cleanly along the lines of different content types, so you might want to look at shifting responsibilities before attempting. Workbench Access sometimes helps in these situations.

5 Translation drift

Once you add translations and internationalization to the mix, you add a whole slew of potential challenges to overcome. One of the biggest is



translation drift, which happens when the main language for a page gets updated but the translations do not. Having more people on the team typically means more content is updated, and things can fall through the cracks. Eventually, the Spanish page, for example, could have content vastly different from the English page.

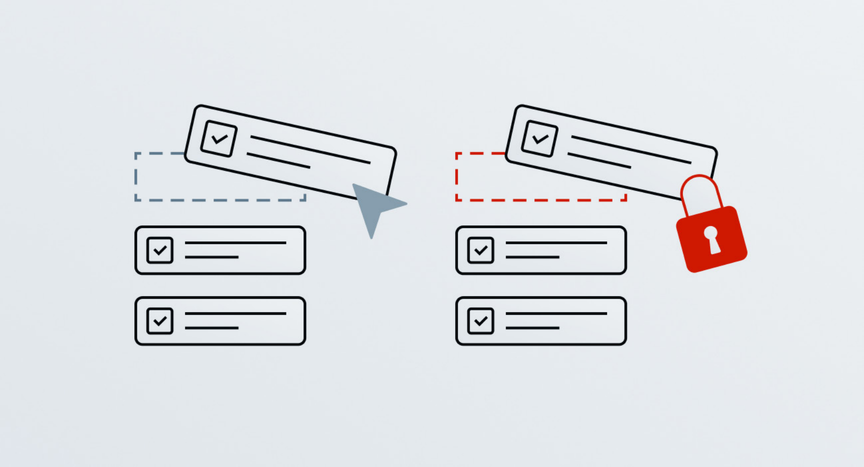
Since translation teams are usually separate from the initial content teams, extra steps are required to ensure everyone is communicating and that all of the dots are connecting.

SOLUTIONS

- **Checklists for content updates.** This is also a good idea for other areas of content publishing, but it helps particularly with translations. Shared checklists inside your project management tools work great. You can also set up some PM tools to auto-generate sub-tasks and assign them whenever you update content.
- **Automated notifications.** Emails, Slack integrations, and more can help keep things like this from falling through the cracks. You don't want to overdo it, though, or else you risk everyone ignoring these notifications.
- **Integrate a proper workflow.** Translations can be part of the workflow, or once the main translation is updated, it can be placed within a separate workflow just for the translation team. Ideally, this happens automatically.

6 Permissions that don’t match the requirements of your team

Out of the box, Drupal comes with “edit mine” or “edit all” permissions and no support for “edit some.” The fact that internal content teams aren’t relevant to a customer-facing website is exacerbated by this. The internal structure of the organization often needs to be hidden from end users but still allow those editors to retain ownership of their content. Combine these two things, and you end up with everyone on the team having the ability to edit anything. The larger the team, the more this is a problem.



The big question: how do you govern access to content within the site when departments or differing teams need to be isolated from one another?

SOLUTIONS

- **Give each team their own content types.** This has the advantage of working with the grain of Drupal. For example, the news team only edits the "news" content type. But if you are not careful, this breaks down quickly. What if multiple teams need an "event" content type? You don't want to create duplicates.
- **Custom access or contrib modules.** There are many contrib modules to test, like the previously mentioned Workbench Access. Drupal is also extensible enough to create your own access solution.
- **Separate instances of Drupal.** Each instance can have the same content types. It keeps both navigation and content ownership away from others in the organization. It also simplifies development and keeps databases smaller. You can even glue these instances into a single URL hierarchy to give the illusion of a single website. This is the "big hammer" solution, and to use it, make sure your problems are nails actually big enough to warrant such a hammer. You can learn more about how the State of Georgia found success with this model of implementation.

Best practices for complex websites with large content teams

Besides specific solutions to specific challenges, there are also some best practices for almost every situation.

1 Start identifying and addressing challenges early

As you may have noticed, many of the solutions presented above are a whole lot easier to implement early rather than late. They aren't impossible to roll out on a current website, but you have to overcome entrenched technology, interests, and habits.

If you are going through a re-platforming or a website upgrade, start thinking about these challenges as soon as possible. Strategy and design expertise can offer valuable insights before you start pouring the concrete. In addition, developers can help you determine feasibility within your budget to help you prioritize the proper technological tools.

Get a good map of the landscape before starting your journey. It's beneficial to run a lot of exercises to help uncover potential problems and to make sure every voice is heard, all before a single line of code has been written.

2 Don't assume every problem has a technological solution

Jumping immediately to a technological solution can sometimes be like adding elaborate locks on a door when there's a gaping hole in the wall right next to it. If your content teams don't trust each other, implementing a strict approval workflow on your website won't solve the problem. It might hide it for a little while, but that's it.

Technology can help implement, enforce, and maintain certain solutions. It is rarely the solution itself.

A well-built form will have the correct help messages and error checks to help prevent editors from making silly mistakes. A well-built form is not a replacement for training. Nor can it replace deep knowledge of how the content model fits together.

One way this has been done is to lock down access to the administration interface of the website until there's confirmation that someone has gone through the documentation and training required. Technology helps enforce this, but documentation still needs to be kept up to date. Editors who haven't logged in for a long time might forget some of the training. There are all sorts of challenges that have to be addressed at an organization's cultural level or at the process level.

3 Consider having content specialists AND content editors

The content needs of organizations have become more complex over time, and content entry has become a professional skill of its own. As content models grow and content reuse becomes a concern, you need to have people who understand how everything connects together. You want content writers writing content. You don't want them worrying about the implications of the content model or asking questions such as:

- How will editing this article affect everywhere else this article is referenced?
- Is it safe to roll back to a previous revision?
- What metadata is appropriate for this content?
- Do we need another taxonomy term, or should we re-use a previous one?

Entering content into modern websites is not just “data entry.” It’s a specialized skill all on its own.

Let the subject matter experts write the content, and let a content entry specialist enter the content.

This usually allows you to have a smaller team directly editing the website, a team specializing in the website itself, which helps solve a lot of challenges upfront. In this scenario, it helps to have a robust preview system for writers and subject matter experts to see their content prior to publication. Great previews provide a natural point of communication between content specialists and the editorial team.

More editors, more problems

You have a large content team because you need to publish and manage a lot of content. But large content teams have challenges that must be overcome so your content pipeline keeps flowing. These challenges can be addressed as long as you don’t jump immediately to a technological solution or don’t immediately assume that adding more people will help fix the problem.

Be prepared to look at the bigger picture first before diving in deep.down the road. The more you listen to and implement feedback, the more trust you build.

This all pays off with greater buy-in and acceptance as your project rolls out and, since many of your stakeholders will already have some familiarity with what you’re doing, easier training.

We’ve helped many organizations overcome challenges similar to yours. If you’re looking for ways to ensure your content team is reaching its full potential, contact us.

About Lullabot

Lullabot is an employee-owned strategy, design, and Drupal development company. As one of the first Drupal agencies, Lullabot is highly recognized for their body of work, authentic approach, and leadership in Drupal innovation, having contributed to more than 150 modules.

- Phone: 1-877-585-5226
- Email: hello@lullabot.com
- Website: www.lullabot.com