Focus on eLearning and Training

Read multiple articles focusing on different aspects of eLearning and training

Communicator

The Institute of Scientific and Technical Communicators
Autumn 2020

Read about MadCap Flare 2020

Discover the latest developer documentation trends

Learn the benefits of HelpXplain from EC Software

Understand the importance of terminology management

Support, customer experience and docs

Are the different parts of your organisation communicating effectively? Joanna Suau shares ways for collaboration with documenting software.

Software documentation, even if perceived as an integral part of a development cycle, is often treated as a must-have to tick all the boxes rather than something that can save development time or money. It is a necessary evil, and many developers hate that they either must document the software themselves, or devote time to talk to a technical writer, so that they in turn can document it for them.

Why is documentation so often seen as an impediment by developers, and how can we, technical communicators, convince them of its value?

Typically, this happens because there are no tangible benefits for developers involved in the development cycle, especially if they still have to answer the same questions over and over again on a forum or through support tickets. Acting as a Subject Matter Expert (SME) to answer questions a technical communicator might have is just an additional chore rather than something that might help them out and alleviate their drudgery.

Why is documentation so often seen as an impediment by developers, and how can we, technical communicators, convince them of its value?

Same goes for a Customer Experience team that has to install the same functionality even multiple times a day for customers that can't do it themselves or for those who just want to feel that personal touch of highly devoted customer care experts.

Documentation as a service

What if documentation was treated as a gateway to communication between customers and a development team? What if documentation was there to help streamline Support or Customer Experience, save money, and make the development cycle seem like a smooth sailing?

Let's talk about specifics. First and foremost, we need to admit that in a typical software organisation there's a discord between teams and that they usually work parallel to each other rather than together as one big mechanism devoted to delivering a unified software experience. Developers develop, Customer Experience onboards[1] new customers or cares for the existing ones, Support resolves issues with the software, while Documentation writes manuals usually for a target audience they have never met or spoken to.

Documenting the undocumentable — internal content ready to use for official guides

Generally speaking, to write a good user guide is to know the target audience. But if you take this a little step further, you could say that to write a usable piece of text is to know what struggles and questions that target audience has.

Often, we, as technical communicators, try to put a user hat on, and write a guide based on what we personally think is challenging when dealing with software we document. However noble the endeavour is, the fact is that we've probably already been influenced one way or another, and are not a good example of a typical user. We've talked to developers and heard them talking to each other, we've read the proof of concept ticket, we've been to a demo meeting, or simply we've been with the organisation for so long that we know all the nooks and crannies of the software we were tasked to describe.

In other words, it's hard to anticipate what a real user could struggle with. You could obviously talk to a customer and observe how they interact with the software, but user testing while better than extrapolating facts based on our gut feeling is complicated. You need to invite users, spend time observing them, and then talk to them about their experience. While useful, user testing is time consuming and costly if you want to opt for a professional service.

Finding the voice of users

The alternative is so simple that it's often overlooked in so many companies. Speaking of users, you already have the perfect way of getting into the heads of your target audience and looking into issues that are most problematic for them. Talk to your Support department to give you access to support tickets, read your organisation's forum, if they have one, and talk to Customer Experience to learn about the biggest challenges with onboarding new customers.

A word of caution here before you jump into making docs out of everything you'll find on a forum or in a ticket. Many questions you'll see are going to be very specific to the particular customer and their own architecture. You probably don't want to write guides on that, unless you wish to promote a specific interesting idea they had, for example, integration with a third party service other customers would benefit from or a really crafty workaround to a commonly shared pain point.

Let's face it, there's no such thing as ideal software.

[1] Onboards / onboarding. The action or process of integrating a new employee into an organisation or familiarising a new customer or client with one's products or services.

Organising your content

A good way to start is to organise the queries topically, and just start with the topic that generated the biggest number of questions/issues.

Though time consuming, evaluate each question and support ticket, maybe assign points as per their relevance and usefulness. Then, all you have to do is to prioritise these, and you have a perfect content roadmap designed according to customer needs. You'll find that in many cases you won't even have to contact Support for additional information, as the entry will be informative enough to adapt to a guide.

Apply the exact same process to Customer Experience. Write down the list of most common queries they encounter to prioritise them, and ask whether they already have any written materials you could use. You'll be surprised at how much content gets created outside of the documentation team: flowcharts, diagrams, quick summaries, steps to follow, etc. Often, other teams consider official guides decorative, while they stick to their own unofficial materials written as a collection of bullet points or loosely gathered thoughts. All you need to do is to adapt this internal content into a customer-facing official guide without the need to schedule a lengthy SME interview.

Once you have a bunch of guides that directly answer most frequently asked customer questions or outline the onboarding process, you can then incorporate them into the support cycle. How you'll do it is up to you and Support to figure out. You may create an FAQ page, incorporate your docs into a chatbot system, or publish the content as standalone guides that you can link to.

Documentation as part of customer experience and support cycle

Documentation seen as the first tier of customer contact is something that is still often frowned upon. It is a complementary service left for customers to dig through at their own leisure. Again, it's nothing personal, it's just that docs are usually a messy jungle of manuals, references, and use cases that only a technical communicator knows where everything is and how to navigate to it. Most developers find it easier to interact with a customer rather than embark on the seemingly never ending journey of searching through documentation.

Also, you'd be surprised at how many people don't know how to search for information — something we are particularly good at and take for granted. This is where you go outside of your comfy shell and start promoting your work. The sky's the limit here. You can gamify your content and make tutorials with a motivating award system, organise your content per audience type to help each group find information most relevant to them, create newsletters which promote content relevant for new patch upgrades, write blog posts promoting your newest documentation and post them on your organisation social media platforms, etc.

This all can be then easily incorporated in a support cycle. As soon as your content is out there, regularly promoted and visible, people start noticing it and will hopefully start using it. Then, written content will gain in value, and therefore it's easier to convince customers to read the information you've sent them, and for developers to send said information.

Many could still argue that it feels impersonal if instead of talking to a person a customer is sent a bunch of links to read through. Just to be clear, I am not talking about sending a list of links without a context every time a customer contacts Support or Customer Experience. I am talking about instances where questions posed by customers can be answered directly with a guide that has been prepared for that occasion. It is not impersonal if it helps resolve their issue in a most efficient way. As a customer myself, I would be highly impressed if a team I contacted to help me was able to predict my struggle and already have some written materials on the matter, saving me time to wait for an engineer to join the discussion and help me out.

You'll be surprised at how much content gets created outside of the documentation team: flowcharts, diagrams, quick summaries, steps to follow, etc.

A skip button

For all those who really don't like resolving problems that way and prefer to talk to a live person, I would think of a way to skip this step and connect directly to a member of the Support team. At the end of the day what you really want to achieve here is to cater for all types of audiences, and we all know a person or two who just prefers to talk to a person rather than reading a guide. The content that you have already written can be reused in this instance to give Support a script they can use with a person who opted out of reading a guide.

Adapting support tickets to include documentation

For the Documentation team to be fully incorporated in the support ticket system there needs to be a really reliable process, as not all tickets/questions will follow the same route. To help with the decision making, I would suggest a checklist or a simple decision flowchart that will help a team decide where to push a new ticket next, dependent on the complexity of each ticket, otherwise you are running into danger of annoying customers with spam or meaningless interactions before their issue is resolved.

Typically, your checklist or a flowchart will list the frequently asked questions or frequently encountered issues for easy reference. You may include links to your most read guides or topics to help decide which team should work on the ticket. As the list will be a live document; make sure somebody owns it, so that it doesn't go stale. With time, some topics will lose relevance because the software changed or simply because there's no need for that workaround anymore.

The workflow needs to be well defined for the system to work smoothly and resemble a healthy automated flow. Document the process and make sure it's easy to follow, without too many steps or variables. If your process is too elaborate, Support or Customer Experience simply won't have time to stick to it, you need to make sure it's friendly for all teams involved, especially for those who you know will quite likely resist the change at first.

Speaking from experience, it's the first few steps that define whether the workflow will crumble down with time. If Documentation is to be the first communication tier, the circumstances in which this step must be skipped is paramount. Otherwise, everything will end up going through Documentation who will soon be swamped with tickets trying to sort them out properly.

Key aspects to consider when thinking about the process flow:

1. Who creates the tickets and owns the process?

Ideally, you'd have the process owner who would oversee the entire support workflow. However, typically, the roles are divided as each team sticks to their own processes and you don't want to shatter all existing ones, you just want to add a helpful layer.

2. Who labels and assigns tickets and what criteria should they use?

Here's the time when your checklist will come in handy. Well-defined tiers will go a long way, allowing all teams to know exactly where the ticket should go first.

3. Who prioritises new tickets?
4. Each team knows their own little processes best, so the best practice is to let each team decide on their own priorities and who works on what. That way you make sure that there's as little disruption to the old process people got used to as possible, especially if the process worked well in the first place.

Sample workflow:

1. Support gets an email and creates a ticket (or a ticket is automatically created on a cross-team board to which Documentation has access).
2. Support evaluates the ticket. If it meets the requirements of the first tier, it's labelled DOC which means it needs to go through the Documentation team first.
3. Documentation team evaluates and prioritises DOC tickets.
4. Documentation contacts the customer to provide initial help. They push it further to Support, if:
5. The ticket proves more tricky and needs more input,
6. The customer requests more information that Documentation is not able to provide.

Why care about customer feedback

The benefits of customer feedback flows both ways. Once customer's feedback is addressed, it makes them feel more valued, while the Documentation team gains additional SMEs that have a unique point of view.

The customers use the software in the real world, so their problems and solutions are driven by real not hypothetical situations. The documentation guides become more to the point and kept up-to-date, as they are now read and commented on more often.

To sum up

As with any new process, incorporating Documentation into Support will be challenging at first, but it's a still rarely used powerful tool that can benefit Support, Documentation and customers themselves.

Being able to talk to real customers can be a very empowering experience to the Documentation team. Writing stops being just for writing sake, technical communicators become more accountable for every word, and writing becomes more real. Additionally, the team gains valuable reviewers whose insight can be really helpful in creating guides.

To make guides great again involve documentation in the support cycle and involve customers in the documentation process!

Related reading

Anstey S (2016) 'Eliminating support tickets' *Communicator,* Winter 2016: 18-20 https://istc.org.uk/index.php?pda_v3_pf=/_pda/2020/03/Comm1612Web.pdf#page=18 (ISTC members only. Login required. Accessed 15 July 2020).

Otero A (2019) 'Social media: an ally in ticket deflection' *Communicator,* Winter 2019: 35-37 https://istc.org.uk/index.php?pda_v3_pf=/_pda/2020/03/Comm1912Web.pdf#page=35 (ISTC members only. Login required.
Accessed 15 July 2020).

Joanna Suau is a technical author that specialises in software documentation. She's a member of the ISTC and has been involved in technical communication for more than five years.
E: joanna.suau@gmail.com
LI: joanna.suau
Medium: @joanna.suau