
CUSTOMER SUPPORT FOR **BOOTSTRAPPERS**



CUSTOMER SERVICE, SIMPLIFIED

CHASE CLEMONS
IAN LANDSMAN



This guide started with one question that bootstrappers like yourself ask me all the time.

"I've got a great product. Now how do I support my customers?"

This short guide will get you started on just that.

Customer support is not a hard concept. Someone has a question or a problem and it's your job to fix it and make them happy. When done right, it can set you apart from your competition. An app that does something can be recreated easily. A rocking support experience is much harder to copy. When customers know they're going to get great support, they'll usually choose you over the competitor.

But if you do it wrong, it can undo everything that you've worked so hard on with your company. Customers will start leaving you behind as they find a company that can provide that support experience they're looking for.

I've split this into three parts - before you launch your product, after you launch, and scaling your team as your product grows. So let's not take up any more of your time and just dive right in.

Part One - Before You Launch Your Product

You've already got a ton on your plate as you get your product ready to launch. Final product tweaks and changes are being made, you're putting in long hours trying to get it ready, and the last thing you want to worry about is the support side of all this. That comes later, right?

Kinda.

Before you launch, you're going to need a few channels for your future customers to get help. Let's focus on the two main ones that you need right now - a clear and concise help site and an easy email setup for questions not on the site itself. The key is to set these up in a way that your customers get answers and you don't get bogged down in support tickets all day. We'll look at other support channels after the launch but right now, focus on these two so you can get back to your product.

First up, the humble help site. A customer's first stop should be your help site where they can find the answer themselves. A solid help site and FAQ will help up to 40% of your customers find the answer without having to contact you.* At five minutes a ticket, those numbers add up.

What should a good help site look like?

Don't get cutesy with names. Call this part of your site the help site. Put it at yoursite.com/help. Then make sure to point yoursite.com/support , yoursite.com/faq , and yoursite.com/questions all to that /help URL.

We're keeping things simple so let's include at least these areas:

- Top FAQs that customers ask.
- Search page so that they can search for answers.

- Contact page so they can ask for help if they can't find the answer.

What should a good help page look like?

When customers are on your help site, they're already confused and in need of help. Don't make it worse with a complicated page. Focus on three things:

1) Clean and simple

Keep the page design clean and simple. Make sure to include screenshots to illustrate what a customer should do. A complicated help page just makes things worse for a confused customer.

2) Short and sweet

Keep paragraphs short and sweet. Lengthy answers should be broken up in easy-to-read sections. Making something more clear does not mean adding more text to the page.

3) Headers and formatting

Use headers and text formatting to make it easier to read. Bolding the right word helps the customer find the answer faster.

What FAQs should I put on there?

Routine and repetitive questions are best handled through self-service. The customer should be able to update their card, change their email address, and things along those lines on their own when they need to. Empowering the customer to do these simple tasks lets them have immediate answers any time day or night.

Beyond that, show the app to a few friends and family members. Right down the questions they have and put those on the site. That'll be enough

to launch with. Once you launch your product, you'll quickly see what other questions needs to make their way onto the help site.

What does a good screenshot look like?

Seems like a no-brainer right? Take a screenshot of what you want to show and throw it up on your help page. Easy.

No.

You've taken care to craft every word on that page. Take the same pride and professionalism in your screenshots.

1) Frame and context

Make the object you're calling attention to obvious. But remember to include it in the context of the page itself. If you focus on just one tiny part, a customer might get confused trying to figure out where on the page you took that screenshot.

2) Use arrows, callouts, etc.

Use them sparingly. A dozen arrows in one screenshot is a tad too much. Highlight one to two things that you want the customer to see.

3) Don't forget the alt text tag

You're going to have people that rely on that alt text tag to "see" the screenshots. Screen-reading software needs the alt text to relay that information to a customer. It'll take a few moments to describe the screenshot but the customer that needs it will love you for it.

Help customers find the info first

You've got to have a solid search tool. No matter how organized things are or how clear cut your FAQs are, customers search for things first. We live in the days of Google search, not a book's index.

Put the search box prominently on the page. Customers should find it as soon as they land on the help site. If it's not readily there, they'll go ahead and send you a support case so you can do the heavy lifting and answer their question. By having the search box first, they'll use it first and find their answer without ever emailing you.

When all else fails, the contact form

If the help site doesn't have their answer, the customer needs a way to get in touch. A simple contact form will do the trick. You need their name, email address, and what they need to ask you. Go ahead and have them pick a category as well. Something along the lines of "I can't login", "I have a billing question", "I have a feature request", etc. Those will come in handy after the launch when you have lots of support cases coming in.

Have that form send you an email to your support email address. And that'll wrap up everything you need for the help site right now.

Examples of great support sites

<http://help.campaignmonitor.com/support>

<http://basecamp.com/help>

<http://wistia.com/doc>

<http://bufferapp.com/faq>

Now that the help site is up, let's focus on customers that need to get in touch with you if their question isn't answered. Your help site has a contact link that's easy to find so that avenue is covered. But watch out for customers that send to generic email addresses like email@yoursite.com, support@yoursite.com, etc. Have email, support, help, questions, etc all redirect to that one support inbox.

You don't need a fancy app to run support. A shared email inbox will do the trick just fine. Gmail's a popular option since it's free but you can use any email tool as long as it has the option for labels and filters with it.

When a customer emails you for help, replying to them is the obvious step. Use labels if you have time. Labels are great for grouping together feature requests, bugs, or any other natural groupings that you come across.

You also want to set up a snippet tool. Personally, I use [Text Expander](#). It's easy to use and really affordable. Text Expander allows you to create "Snippets", which are bits of text you can insert into an email with only a few keystrokes. It's easy to organize snippets into folders and extend them for all sorts of use cases.

Your snippets aren't locked into your support app so you can always take them with you if you change. You can go with other apps if you'd like but for this one, we'll be focusing on Text Expander.

How much you use the snippet tool is up to you. I use it for any phrase that might be repeated. For instance, here's the ones I've got set up and use on almost every email:

Hi {{name}}!

This one's for opening the email. People have a name – use it. It goes a long way in showing that you know who they are and that you're there to help them personally.

If you have any other questions, just let me know and I'll be happy to help. And have an awesome {{today}}!

-Chase

This one's for closing the email. Short and sweet but it lets your customer know you're always there for them. **Queue that song from Friends.**

Those two snippets alone will save you a ton of time. You can add to them as you need to.

The last thing I'll mention is to watch the wording you use. Don't use corporate buzzwords or other lingo when talking with customers. Two big email types you'll probably see from customers are bugs and feature requests.

Product launches always turn up a few bugs and other troubles. If they run into a bug, just say you're sorry like this:

I'm so sorry for that trouble! I found that bug and just went live with a fix for it. It should be good to go.

For feature requests, you'll want to be nice but make sure to not promise anything.

Thanks for sharing that idea! I can definitely see where that would help out. I can't make any promises that it'll make it into the app but I'll keep it in mind.

Combine those bug and feature request emails with the opening and closing from before to have a solid email reply.

Part Two - After You Launch Your Product

Your product's launched! Let's just take a moment and celebrate before we jump back into the support world.

Alright - your two main support channels are doing fine so far. Things are starting to calm down from the launch. It's a good time to be thinking about other support channels.

As more customers start using your product, you're going to get a ton of pressure to expand your support channels. You started off with the help site and email but now customers want support through Twitter, Facebook, phone, SMS, live chat, and who knows what else.

Resist the urge to cover everything. When you cover too many support channels, you dilute your effectiveness. You're a bootstrapper so you can't be everywhere all the time.

Instead, become a master of a few specific support channels. You've nailed the cornerstone of support with a help site and email. From there, you could expand into Facebook or Twitter, since lots of your customers are already there. Or maybe you want to invest energy into a live chat tool. The specific mix is up to you – just make sure you become a master of a few rather than a novice at them all.

All the different options

Let's look at each one. I'm a pro/cons list kind of guy so that's how we'll break these down. They're in the order that I think you should focus on expanding into.

Twitter

Pros: 140 character limit means it's perfect for short questions and answers. For easy answers, a simple yes or no and a link to a help site works great. Your product/company is probably on Twitter already so just start answering questions there too.

Cons: 140 characters isn't enough for all questions. That's when you just send them your support email address so they can send you an email. Twitter's also a very public place where others can jump in on a thread. If it's a Tweet from an angry customer, it can explode into a public nightmare real quick.

Facebook

Pros: Lots of people use it already. It's easy for them to find your page and interact with you. There's no limit to the characters like Twitter so they can have longer conversations with you.

Cons: Similar to Twitter, it's a very public forum. If something goes wrong, it can become that publicity nightmare.

Live Chat

Pros: Customers love live chat. They can instantly interact with someone, almost like a phone conversation. When offered, customers jump to this channel. From the support side, it's nice that it eliminates back-and-forth email replies that can drag on for a day or more.

Cons: It's an interruption machine. Just when you get into your groove on something else, you'll get a popup from a customer wanting to chat. It's also people intensive. You can handle dozens of emails in a 10 minute span. With live chat, you can only juggle maybe three chats at once. That means to scale it, you have to bring in more support reps.

Community Forums

Pros: It helps take some of the support pressure off you. Customers can help other customers with questions. They can share ideas between themselves.

Cons: The pros only happen in a utopian world. It's equivalent to outsourcing your customer support to someone else. Customers aren't even getting paid to answer those questions from other customers. That means they don't have to worry about responding fast or in a clear way. You lose control of the support experience.

Set those expectations.

With all of these support channels, the key is setting expectations with your response times. I want you to say this next part with me.

It's okay to take off holidays. It's okay to have open hours.

In the world of out-supporting each other, many teams turn to a 24/7/365 support availability.

“See!” they cry. “We’re here all the time via email, phone, Twitter, Facebook, snail mail, LinkedIn, and that brand new app forum that just started yesterday. We’re here!”

You're one person right now. It's okay to set open/closed times. It's okay to put on your help site that you're one person answering emails so it might take a bit longer, especially since it's right after a launch.

As long as customers know what to expect, they'll be fine with it. At the end of the day, let work happen during work hours and go spend time with family and friends outside of that.

Prioritize like crazy

After a weekend of family time, a full support queue can be daunting. Where do you even start?

For that, we turn to prioritizing and some automatic awesomeness.

Remember that support form you created for the help site? We're going to use those contact categories to help set up a triage of tickets. We're going to go with a scale of one to five with a priority of one at the top of your list.

Priority 1 -->> “I can't login.”

Priority 2 -->> “I have a billing question.”

Priority 3 -->> “I think something's broken.”

Priority 4 -->> “Other” and emails not submitted through the form.

Priority 5 -->> "I have a feature request."

Have your form submit to your email address and apply a label for one of those categories. To catch the ones that skip that form and directly email you, set up a filter to catch words like "login" to match with a priority one, "unknown charge" to match with a priority two, and so on. As you get more emails, you'll see other patterns that you can match to those labels.

Once you have those set up, you can work on the priority one stuff to start and then make your way down the queue.

One last thing - keep an eye on patterns. Are you seeing lots of people asking for password resets? Maybe your design should be tweaked to make it easier to find. One or two people having trouble may not be anything but multiple people running into the same situation means something is off.

Scheduling support time

So when should all this support answering happen? You'll know your day better than anyone else. Your schedule is unique to you. But since you asked for advice...

Set aside a few time slots each day to handle emails. When you're first finding a flow, try one in the morning, another at lunch, and a final one later that afternoon. Clear out the queue during those slots. That way, no one is waiting more than a few hours before they get a reply.

Think back to your hours before the launch. Remember how you'd take a break for lunch, maybe one in the afternoon? The timing of those breaks will give you an idea for when to slot your support time in.

Just remember - your schedule's unique to you. Find the time that works best with your daily flow.

Part Three - Scaling the Team

So far, we've looked at the tech side of things you can do for bootstrapping the support side of your product. Let's shift into looking at the human side a bit more. After all, all the technology in the world won't make up for having the wrong people on your support team.

When a customer emails you, you need to have an actual human respond. As more emails come in, you'll need more people to help reply to those customers. When you reach your limit, call on the rest of your team to pitch in and help.

Lots of small teams are taking this all-company support approach. Bringing on your partner, designer, programmer, etc. to work support cases is the same as hiring on another support-specific person, only at no extra cost.

There's a couple of ways to start with all-company support like this. You could have each person take a support rotation for a few hours or half-day. Or you could have each person answer a set amount of support tickets a day.

The last thing I'll mention here is to **keep everyone fluent with your product**. You want everyone on your team able to field any question they might get from a customer. When your team is fluent in all areas of your product, they'll be able to provide a better experience to your customers. There's no weak link then that might break under certain situations.

Of course, this all-team support will only help with the growing support load for so long. At some point, you'll need to hire more support reps. But when?

A good rule of thumb here is to hire when it hurts. This lets you feel the pain of being short staffed. You'll have a clear picture of what you need to hire a person to do instead of just guessing at what they should do.

Start looking for a dedicated support rep when:

- You spend more than half your work day working support.
- Your response time is more than 24 hours to a customer.

Don't freak out the first time either of these happen. It could just be a slight bump in your support load. When your support load is constantly heavy like this, go ahead and start looking for that dedicated person.

Wrap-up

It's easy to be bad at support.

Companies excel at that all the time.

Why not be the one that gets it right?

With this mindset, and great tools, support can truly become your competitive advantage.