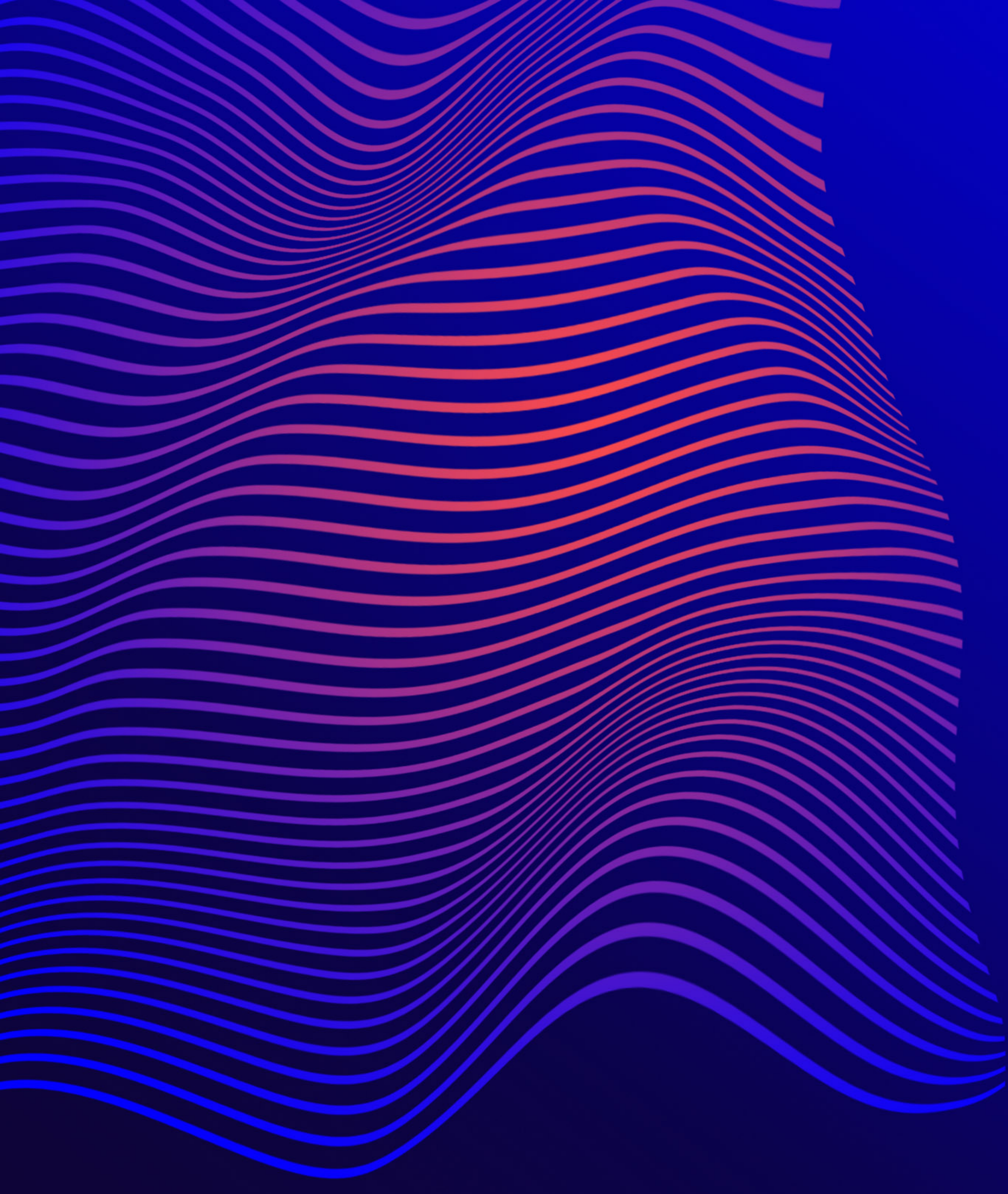


Intercom on Customer Support

—2nd edition

TABLE OF CONTENTS

Intercom on Customer Support

Sharing what we've learned about supporting customers at scale, while retaining the power of personal conversations and targeted content.

Intercom's software enables internet businesses to see who is using their product, and makes it easy for teams to support those customers using a collaborative team inbox, which is designed for working fast while staying personal. Find out more.

We regularly share our thoughts on customer support, product management, design, startups, and the business of software on our blog.

Thanks to Raechel Lambert, Tom Richards and Mark Ryan for all the feedback and domain knowledge. Thanks also to Bobby Pinero, our Director of Finance and Analytics, for explaining the finer points of COGS and other SaaS accounting terms.

Cover and chapter illustrations: Tim Gilligan

Copy editing and proofreading: Geoffrey Keating, Nathalie Marquez Courtney and Adam Risman.

ISBN 978-0-9861392-5-3

Editor’s note

Welcome to the 2nd Edition of *Intercom on Customer Support*. A lot has happened since we published the first edition of this title back in October 2015.

Back then we were only beginning to think about how to approach the job of customer self-service. Intercom was, and still is, synonymous with making personal, relevant conversations the bedrock of a great support experience for your customers. We believe great support is more about conversations than tickets.

But we knew a great support experience can also be delivered when people are able to help themselves. Everyone wins if customers can quickly access answers to common or easily resolved issues. We’ve addressed self-service support with the release of Intercom Educate: our take on what a knowledge base should be in the messaging era.

It would have been remiss of us not to update this book at the same time. A version of *Intercom on Customer Support* without advice on how to create and manage great help content, would only be telling half the story. We know creating quality content is hard – that’s why there’s always a temptation to put the hard work into content marketing and leave help content as an afterthought to a product or feature launch. So we’ve added a brand new chapter on creating quality help content with lots of practical advice on knowing what to write, how to write it, and how to maintain it.

We also took the opportunity to give the rest of the content a bit of a spring clean and broke it down into more easily digestible sections. We hope the result will be of continued use as you grow and scale your support teams.

- John Collins, Managing Editor at Intercom.

Foreword

There are many books about the tactical aspects of doing customer support for a software product. This isn't one of them. This isn't a playbook. It won't guide you through reducing your ticket count, suggest an escalation policy for your support reps, or prescribe solutions with zero knowledge of your company.

Should you support free customers? How long can you make a customer wait? What tone should you take in your responses? How should you scale your support team?

The answer to any of these age-old questions is always the same: it depends.

What it depends on is what we cover in this book. *Intercom on Customer Support* explains how you should think about support and offers guidelines and frameworks to inform every decision you're likely to make.

In the not too distant past, customer support was seen as a hassle, a cost that had to be borne but which was really just a tax on success. If it wasn't for those pesky customers with problems, profits would be through the roof, right?

Luckily that adversarial view of customers is on the wane and progressive companies consider a customer-centric culture a core value. Putting your customers front and center is essential to customer retention, transforms customers into advocates for your business and delivers a competitive advantage.

Fundamentally we believe that to grow a great product company you need:

- happy customers
- who are highly engaged
- stick around
- and are continuously providing feedback to improve the product

And each of those factors - customer happiness, engagement, loyalty and feedback - can be influenced by support more than any other function of your business.

The thing about support is that if you don't get a handle on it, it'll get a handle on you. Just because something is fixed this quarter doesn't mean it's going to stay fixed. That's why we begin this book by looking at how you build and grow a support team. We go right back to when the founding team is providing support from their own inbox and consider the challenges you face as things scale.

In Chapter 2, we'll share some of the things we've learned about hiring teammates to work in support - how to do it, when to do it and why it doesn't fix everything. Chapter 3 is about the choices you'll make on what level of support to give your customers and Chapter 4 dives deeper into how to implement those choices.

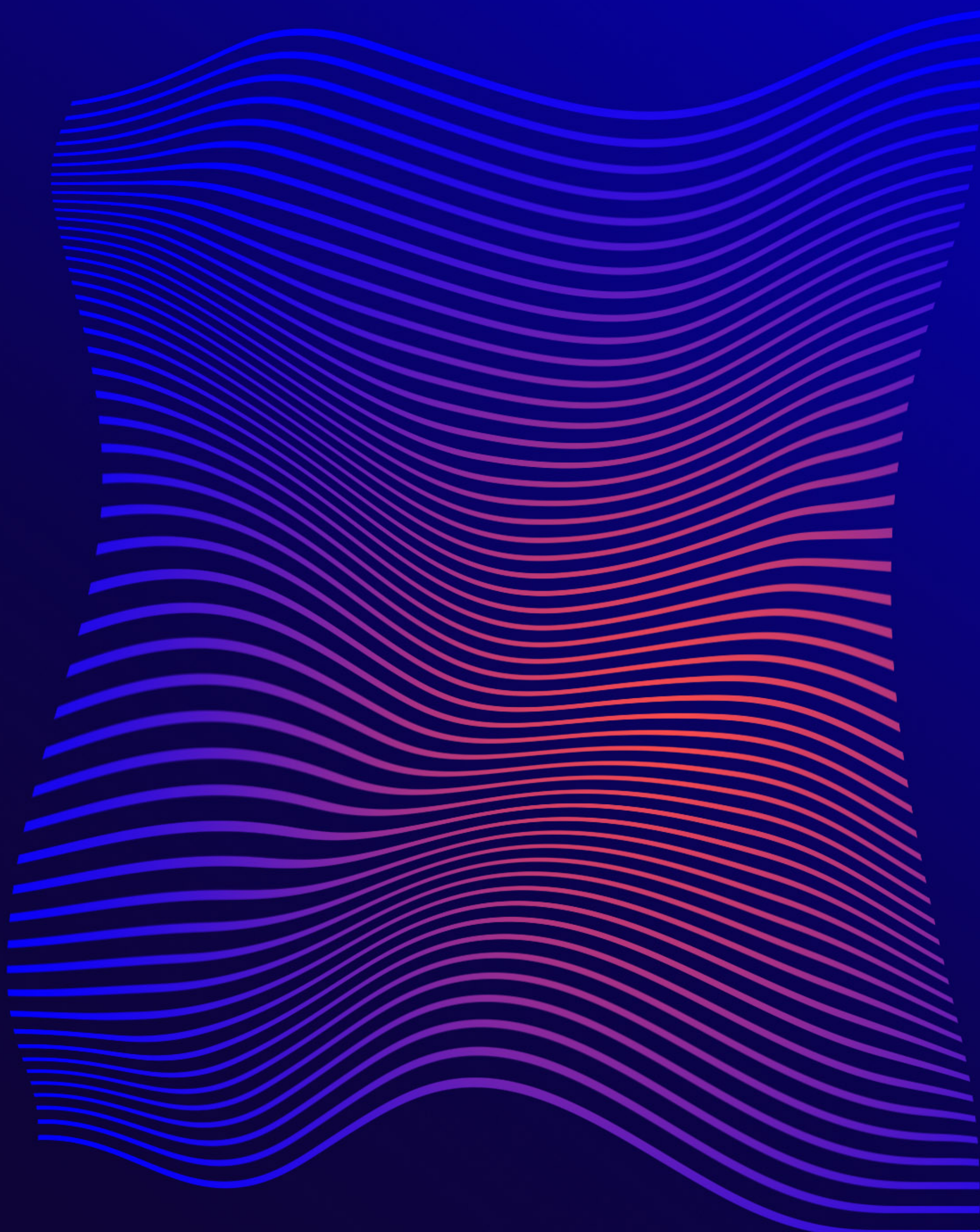
Great support doesn't always involve a conversation with your support team, so in Chapter 5 we cover best practices for helping customers help themselves. The book finishes with an often-neglected aspect of your support operation – how best it can interact with the rest of your business, in particular the product and sales teams.

Our hope is you find our lessons, guidelines and frameworks beneficial as you start, scale or fine-tune your support organization. We'd love to hear your feedback – drop us an email or get in touch through Intercom. And be sure to check out our blog, where we regularly write about customer support and other issues relevant to growing software businesses.

- Des Traynor, Chief Strategy Officer and Co-founder of Intercom

Support at the beginning

Chapter / 01



Ask any software company if they provide great support and they will universally answer “yes”. If you ask their customers if their support is great, you’ll rarely hear similar. Why is that?

Every company sets out to deliver fantastic support, so why do so many fail? Typically what happens is this. The founders care deeply about delivering a great customer experience. That’s what they set out to build for their users – a better way of doing things that’s more efficient or enjoyable than existing solutions. However, with success comes scale…and support doesn’t scale well. Put simply, customers emailing the founder every time they’ve found a bug is fine for six users, not so much for 600, and it’s really broken at 6,000.

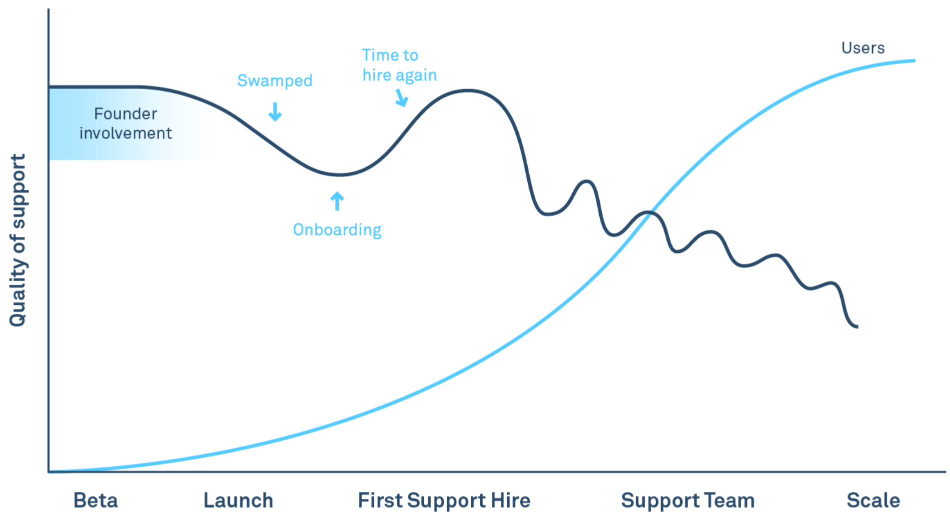
As your company grows, adds staff and layers of management, an interesting thing happens. The people who are most empowered to improve the customer experience become most distant from it. Different teams spring up and start work on different parts of the application(s) with the result that information gets localized and decentralized. It becomes doubly important the support team has access to those key people. If you want to deliver exceptional support to your customers it is essential your support team is fully resourced, has access to senior leadership, and is respected by the rest of your organization.

Misunderstandings, confusion or product complexity are the common reasons people will need support using your product. For example, if people think your project management tool should track time and go looking for your time tracking tool, they will contact support or hit your help site when they can’t find it. That could be deemed a failure of marketing. If people can find, but can’t configure it, that could be deemed a failure of product design. All of these issues lead to an ever increasing demand for support as your company gets bigger and product usage grows. The gravitational force of scale on all product companies means there is a huge temptation to fall back on quick answers and hacky solutions, to delay, outsource or over-triage your support. Over time this means the quality, personalization and responsiveness of support tends to decrease.

Where to start with support

Customer support comes in different shapes and sizes. It's important to think about how your support has evolved, and will evolve, over time. This first section will describe each of the key stages that support grows through within a rapidly growing company.

You start with a founding team, which typically has no deep understanding of how to deliver a great customer experience through support, attempting to scale an organization to do just that. With little understanding of the metrics that matter most, they're likely to repeat the same mistakes faced by other early-stage startups.



Everyone starts in the right place

When you are launching a new product or you are in private beta, you care for every customer as if they were your last. You're so concerned about maximizing your probability of success that any customer feeling annoyed or disappointed by their experience hurts you deeply. You'll do all sorts of crazy unscalable things that people don't normally do. You'll reach out to them personally. You'll give people your personal email address. You'll reply at all hours of the day and night. All to maximize your chances of success when your product is in this delicate, fragile, infant stage.

It's also why every product starts off with great support - customers are literally talking with the founders of potentially billion dollar companies on a one-on-one basis. So where does it all go wrong?

If your product gets any modicum of traction, you can be guaranteed your support volume will grow. This happens even if you have the most perfect product in the world with zero flaws, confusions or bugs. People like to ask questions and learn more about your product; best practices, the future roadmap, best ways to use a feature, why you don't have a feature, how you compare to competitors, etc. All of these questions pile on top of each other, being asked by the ever increasing audience of people who use your product, and before you know it you have a full time job on your hands.

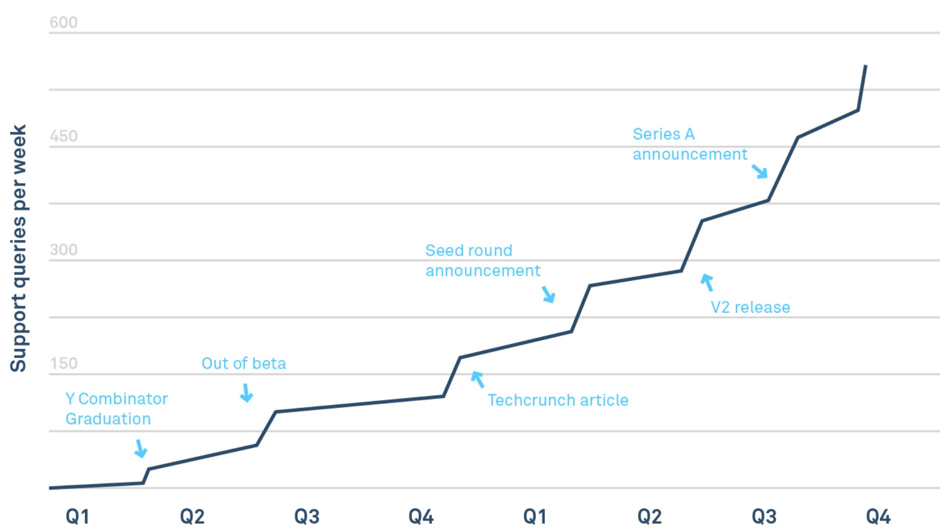
In reality your problems started much earlier than that. As a founder or early stage employee at a startup you have many jobs. You're likely the marketer, the product manager, and the developer. You're probably even the person who empties the trash cans in the office. By the time you realize support is a full time job it's probably too late - you're either providing poor support or you've sacrificed all your other commitments to do it properly.

At what level of workload it makes sense to hire someone dedicated to handle your support depends on the nature of your business, the kind of support you want to provide, and the complexity of your product. But here are some things to consider.

While a founder continues to handle support the conversations are not going to be quick or lightweight - you'll possibly know each customer and you'll also want to get the most feedback from every customer contact. And of course customers are going to want to make the most of a chat with the founder.

Hiring a dedicated support person early on buys everyone else a lot of focus. Before that hire designers and engineers are going to be regularly interrupted to answer questions. But doesn't it make more sense for them to be getting customer feedback through support rather than answering random questions for five minutes every hour? Your first support hire owns the random stuff and ensures it is dealt with in an orderly fashion.

You need to look at the trajectory you're on and allow time for hiring and onboarding - you won't hire straight away, and even when you do find the right person they won't leave their current job instantly. Best case you're a month away from your new hire being able to provide support. During that time what you thought was half a role has turned into a full-time job.



When you know support is taking a significant amount of the founders' time, consider hiring a support person whose role can be augmented with tasks and responsibilities that leverage the same skill set. Customer outreach, customer success, and retention are usually a good fit.

The moment you hire your first support person and onboard them, you are now one degree removed from the customer experience. You need to bridge this gap to ensure you don't lose touch. Losing touch looks like multiple customers being angry about the same issue and you not hearing about it, frequently occurring questions you are not aware of, the product team not hearing about a negative response to a feature release, etc.

Your first hire is also a great time for you to start getting a real handle on basic metrics such as time to respond and agent workload. At this point you should also start to consider things like tone of voice, types of support you'll offer, things you'll never do, etc.

Your first support hire should be responsible for your help content (docs, FAQs, etc) too. Sure its likely to be very thin at this stage, but putting some shape on it early on gets the support team in the mindset of thinking there are ways of reducing the support overhead. You don't want to create a culture that sees self-service content as *only* reactive and a way to deflect conversations.

Creating a help site shouldn't be some major heavyweight project with milestones, project owners and launch dates. You can begin by crafting replies to commonly asked questions which you can quickly reply to conversations with. This can then become the basis of a published FAQ, which in turn could be the precursor to a full blown help site.

Ultimately the copywriting and production of your help content may pass to marketing or some other team but you should always ensure there is a healthy pipeline of requests for changes/additions to it coming from support. Your product teams should also be involved in creating new help content e.g. proactively adding pages to coincide with new feature releases. If your help content is only reactive it will never have the impact on your support volumes that you were hoping for.

From one to many

Now that support is out of your hands you need a healthy set of metrics to help you predict and plan your hiring so it's proportional with the evolution of your customer base. Note these are not directly proportional. There are lots of tools and techniques that will help disconnect them somewhat, which we will cover later.

Put simply you need to think about hiring in advance of demand. As Henry Ford famously said, "If you need a machine and don't buy it, then you will ultimately find that you have paid for it and don't have it." The same is true for support. If you need another support rep and don't hire them you'll pay the full cost of hiring them but still won't have them.

The metrics you need to look at here are:

- **Response time.** It's a loose proxy for customer experience.
- **Response quality.** Hopefully at this stage, as a founder, you can still eyeball what is going out and coming in, to make sure your customers are happy. If not, there are many tools that make it easy to survey the people who have contacted support to see if they've had a good experience.
- **Workload per agent.** All support teams are vulnerable to burnout.

> Imagine having one really enthusiastic, passionate customer support person dealing with 50-60 conversations with customers per day. Consistently, without reprieve, and with the workload only growing. This is not scalable, not sustainable and a bad idea, particularly with a good agent. You will lose them and you will be back to where you started as a single founder, except you will be much further along the growth curve with many more support requests to handle.

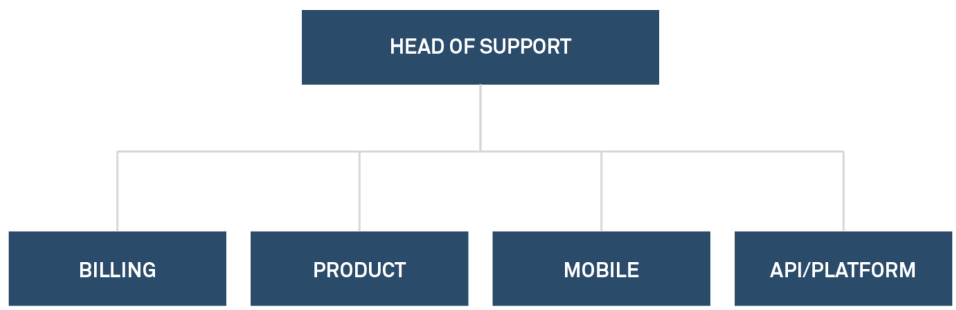
It's essential you have a grasp on your metrics so you know when it's time to hire again. As you start to grow your team it's the basis of your support structure. Hires two, three, and four will remove you further and further from your customers. It is at this point you are going to have to appoint someone as the voice of the customer who can communicate directly with the founders, product teams and senior leadership.

And then you hit scale

When you are into the upper echelons of active users, e.g. 5-10,000 daily active users, you'll have a range of topics that you have to deal with. There are two broad approaches you can adopt to deal with the growth in demand - developing specialists (scaling vertically) or have some sort of tiering (scaling horizontally).

If your product has evolved such that there is a large amount of specific domain knowledge needed to support it, your best response will be from domain experts. The way to breed domain expertise is by restricting their world view. Put another way, the person who knows everything about how your API works shouldn't also know the ins and outs of your refund policy. The risk associated with a vertical approach is that you have a potential single point of failure – what happens when the API specialist goes on holidays, is sick, or isn't available?

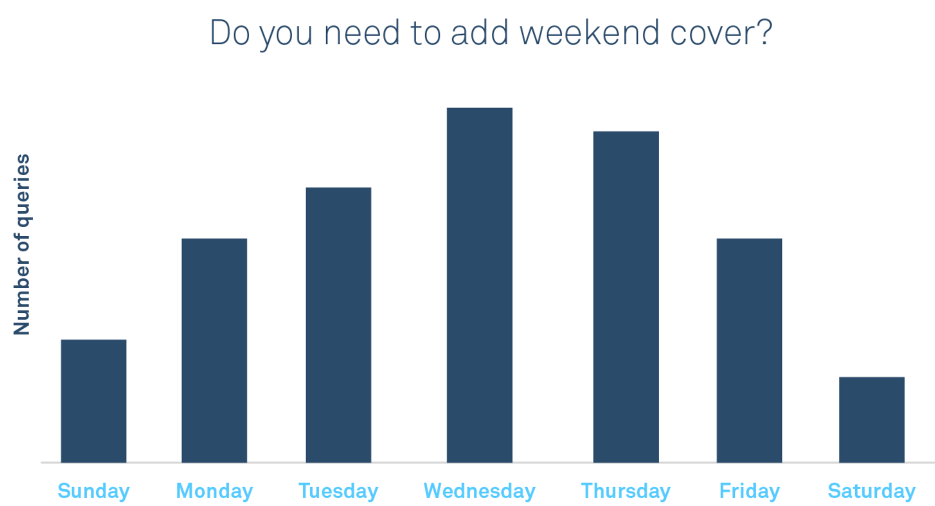
Example of a vertical support organization



The alternative is to tier your support. You have some people who will deal with a large volume of requests about features, triage problems, etc. At the next level you have more technical staff who can do a deep dive to investigate problems or maybe even carry out simple product fixes. There's definitely a trade-off between the number of handoffs a customer goes through and customer satisfaction so your tiered approach may be as simple as having reps who know how to use the product extremely well and engineers who understand the code that runs underneath it.

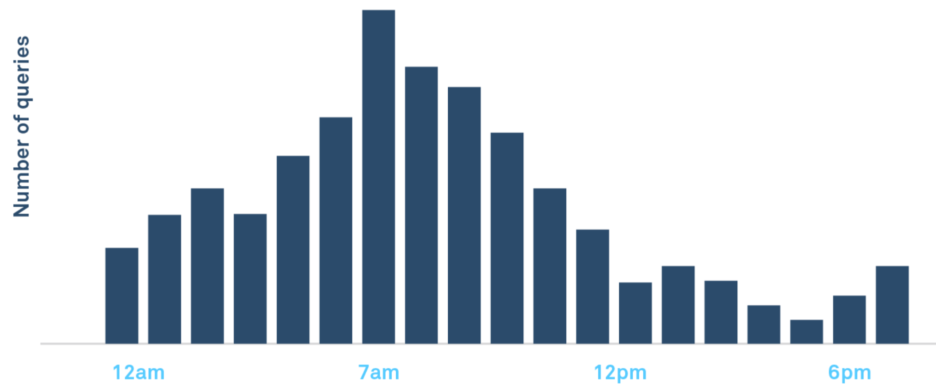
At Intercom we have both non-technical and technical people on the team. This means we help customers understand and use the product better and also deeply investigate, report, and sometimes fix bugs with the product.

As you scale you are also going to need to consider whether you offer 24/7 coverage. Providing 24/7 coverage means you are going to have people in different parts of the world, in disconnected offices. You'll also have people who work weekends, possibly exclusively, which means these people are fully disconnected from the majority of the company. Even the most extreme B2B products need weekend coverage. You also probably need to start thinking about multi-lingual support. All your decisions around when and where to provide support should be informed by your metrics.



If you are seeing a graph like this it's time to think about how you might be able to cover weekends.

Do you need cover in additional time zones?



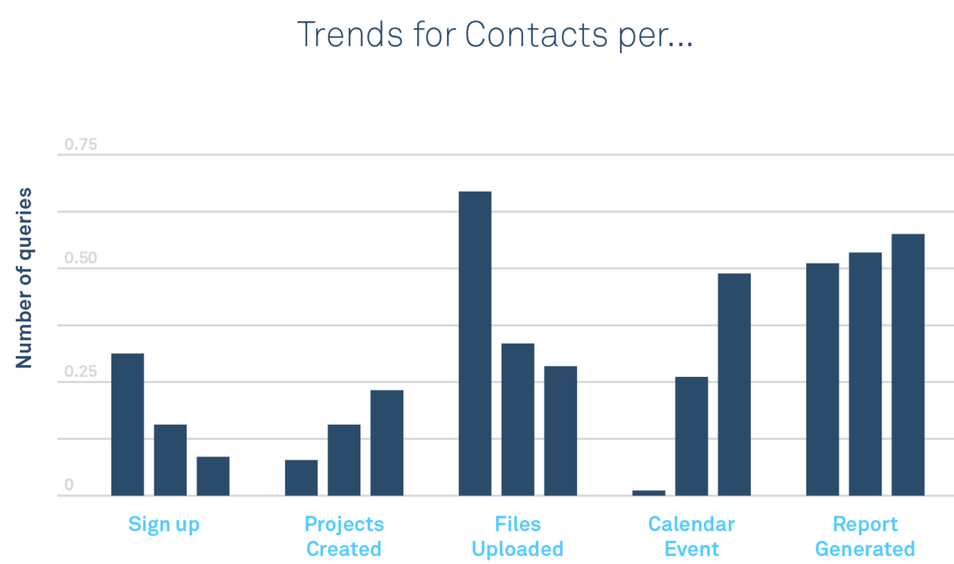
If you are seeing spikes in activity during hours it's hard to service from your home time zone, maybe it's time for overseas cover.

At scale you are out of excuses. Team structure, emergency response policies, processes, your support org's tone and voice, how support is specified, etc, all need to be working well and robustly. Your product is no longer this fun little thing being run by nine people from a cozy office in Idaho. When customers start paying for your product, they expect you to be able to support them when they really need it. You need to think of everything.

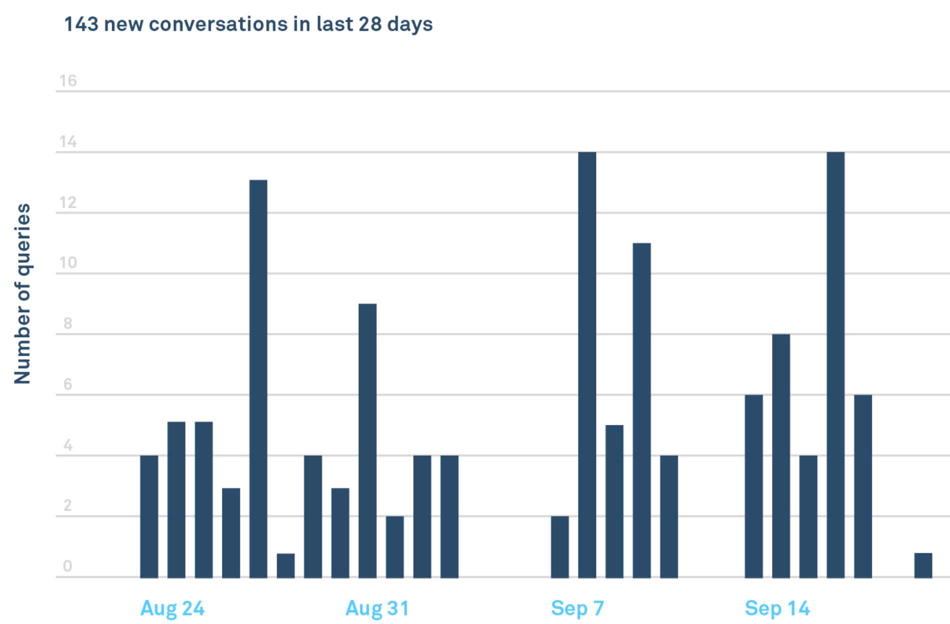
It's worth adding that if the head of support doesn't have a direct connection to the highest level of leadership in the company, then the company is by definition out of touch with customers. You can produce all the research reports and run all the focus groups you want, but if the highest escalation point for customer complaints is more than one degree removed from senior leadership, you are out of touch.

Measuring support

The only way you'll be able to understand how well you're supporting your customers is if you are tracking, and acting on, the right data. Metrics are also indispensable for planning hiring and ensuring your team has the resources they need to succeed. As with all things though, it's important to keep them in perspective. Your job is supporting your customers. Metrics can help you do that but should never be used to justify acting in a way that doesn't benefit your customers e.g. a company looking at the data below could improve the metrics by disabling its report generation tool but that wouldn't benefit customers.



Below are the metrics we feel are most important for support teams to track. It's not an exhaustive list but these are the key numbers we've tracked at Intercom as we have scaled our support organization. Looking at these stats over a variety of timescales and in different groupings highlights patterns you might otherwise miss. Our experience is that being able to view the data over one-week, one-month, and six-month timescales is generally adequate. Viewing it in a simple time series and also grouped by day of the week and hour of the day helps you quickly identify patterns.



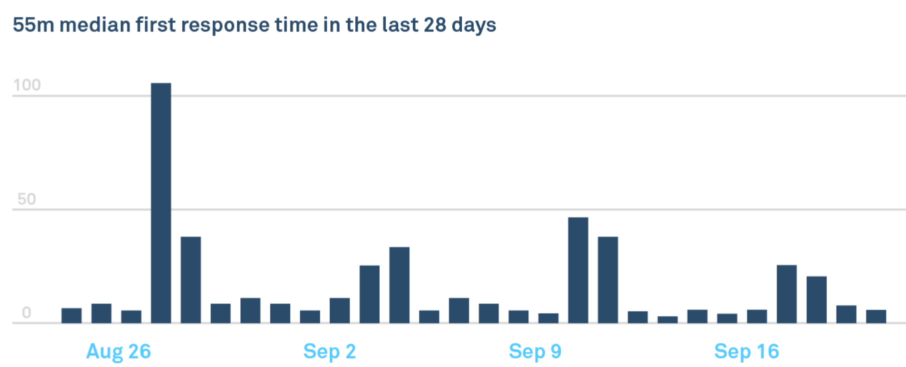
Total conversations. A count of all the interactions you had with your customers. This is a good place to start when building a hiring plan and gives you a good idea of how much work is on your team's collective plate.

How busy is my team?

TEAM MEMBER	REPLIES SENT	CONVERSATIONS CLOSED
Tammy Martinez	128	105
Robert Kim	54	47
Brian Gordon	108	89
Lauren Morris	64	17
Michelle Hu	47	34

Conversations per teammate. At Intercom, we track the number of conversations each teammate handles (more traditional customer support operations call them cases). We also have benchmarks for the number of conversations per day we expect to be handled for the different types of roles within the team. These benchmarks give everyone a clear idea of what is expected of them. However, there are many reasons an individual might fall short of a given benchmark e.g. getting wrapped up in solving a particularly complex issue for a customer, so this stat is always used as a discussion-starter instead of a quota. Conversations per teammate also shows when and where the team is being overworked and needs backup.

Conversations per teammate per day. This stat is the cornerstone of our recruitment planning. Generating the metric is fairly simple - just divide the total conversations your team handles by the number of people on the team by 7 days (or 5 if you don't offer weekend support). Deciding on what's a "good" value isn't as straightforward. It will be different for every business, every product, and every team. At Intercom, roughly 35 conversations per teammate per day is what we shoot for. We arrived at it after looking at a lot of historic data and honestly discussing our workload within the team. It's important to note we regularly revisit and update this figure as the team and our workload changes.



Response time. From a customer point of view, lower response time is always better. But, depending on how you've defined your support, response time is often the first place you make trade-offs. For example, you may decide free customers won't get a response until all queries from your paid customers are dealt with. If you go down that road don't be surprised if you see free to paid conversion rates drop off. We'll get into the detail on supporting free customers shortly.

A significant outage that generates lots of customer contact, as per the graph above, can also skew your metrics. It's also important to remember that time to first response can be gamed easily by your team - it can be as simple as answering lots of calls and hanging up immediately or copying and pasting canned responses into messages. For this reason alone, making it the primary target for individual performance isn't wise. Both the average (mean) and the median - which limits the effect of outliers in the data - are a good start. A very useful additional data point is to look at the 90th percentile value. This is the longest wait time for 90% of your customers that get in touch.

Grouping your response time stats by day of the week and hour of the day is especially important. Spotting times of the day or week that you perform badly is a clear indicator that you might need to organize your team differently e.g. staggered shifts or hire people for the hours you are not covering adequately.

Customer satisfaction. Total number of conversations and time to first response give you a good quantitative idea of how much work you're handling and how quickly you're handling it. Unfortunately, it doesn't give you any idea about the quality of the work you're doing. There are many ways to measure customer satisfaction but making it ubiquitous and extremely lightweight is key to getting a large number of users to respond. A popular approach is the one pioneered by Basecamp. They added a simple question to the signature of all their support emails asking how they did and giving the user just three simple links to click (Good, OK, Bad) in order to register their response. There are also many products that allow you append a simple Net Promoter Score (NPS) survey to the bottom of emails.

How support fits in your business

There is no escaping the fact that supporting your customers costs money. Sure, you want to provide the highest quality support possible but the more you spend on support, the more you are going to eat into your profits (or increase your losses). To get a better handle on how support fits into the broader business, we need to look at financial models. Don't skip to the next section; we'll keep it pretty simple!

The costs associated with providing support to your customers are included alongside hosting and some other related expenses in a figure known as Cost of Goods Sold, aka COGS. Unlike traditional industries such as manufacturing, where COGS are made up of physical costs incurred in selling your product, SaaS businesses are treated as services businesses, so COGS aligns with the costs of providing that service to your customers. COGS includes:

- Hosting
- Other third-party costs e.g. content delivery networks, software, etc
- Customer onboarding costs
- Support team costs (including personnel)

Hosting and other infrastructure costs are relatively fixed - each additional customer doesn't move the needle - so support costs are the easiest way to increase or reduce COGS.

Here's a simple way to think about COGS in the SaaS world. If you stopped shipping new product in the morning, stopped signing up new customers, but continued to serve your existing ones, what costs would be incurred in keeping the lights on? That's COGS. It's an important number and is key to telling you if your business fundamentals are sound.

If you want to go deeper on SaaS metrics, David Skok has written the definitive blog posts and he very clearly makes the point that COGS (and by proxy support) is a key component of profitability for SaaS companies.

$$\text{Profitability} = \text{Revenue} - \text{Expenses} - \text{COGS}$$

There are two implications of this. If you want healthy financials your support team is always going to be busy. No matter how much you value your customers, you can't cut an open-ended check for support if you want to build a viable business. On the flip side, cutting support to the bone is a tempting strategy to increase profits. That might work in the short term, but bad support will lead to increased churn which will erode your profits just as quickly as you increased them.

So, how can you better understand your ability to spend on support? Economies of scale, your position in the market, and your pricing should influence the way you think about spending on support. Here are a few examples.

Pricing is probably the simplest of them all. Why can Apple afford to provide much better support than Dell? It's price. A similarly specced computer from Apple and Dell will not be priced the same and may even vary by orders of magnitude. Put simply: the more you can charge for your product, the more support you can afford to provide. Figuring out how to charge more, that's the hard part. In SaaS specifically, the more you can charge while not negatively impacting your retention rate proportionally, the more you can spend on support.

Economies of scale on hosting and other infrastructure also have an impact. Imagine you are running a fictitious product management application that sells access to its product on a per seat basis. Each additional seat sold costs you almost nothing from an infrastructure perspective. Assuming you're using third parties like Amazon Web Services and reserving instances along the way, the more seats you sell, the lower the marginal costs to host each additional seat. As hosting costs per seat decrease, you can increase your support spend per seat while maintaining the same gross profit margin.

Your place in the market also dictates how you're able to support your customers. It's just a simple fact that larger, more valuable customers will churn less. It certainly is more expensive and difficult to land them (often with involvement from the sales team, product and engineering teams to answer/build specific requests, your CEO, etc.), but once you've landed them they're much less likely to quit. Their switching costs are higher. They've invested more of their time, money, resources, and reputation to adopt your product.

In contrast, smaller customers adopt the product quickly and cost much less to acquire. However, it's much easier for them to quit. Their switching costs are much lower, and it's also statistically more likely that they go out of business. This natural difference in retention across markets will completely change your lifetime value (LTV) equation and dictate how much you can spend to support customers. Said another way, an hour spent with a larger, higher-retention customer makes up a much smaller portion of that customer's LTV than it does with a smaller customer. If you want to treat all customers as equal go ahead and do it. But be clear about the trade-offs you are making.

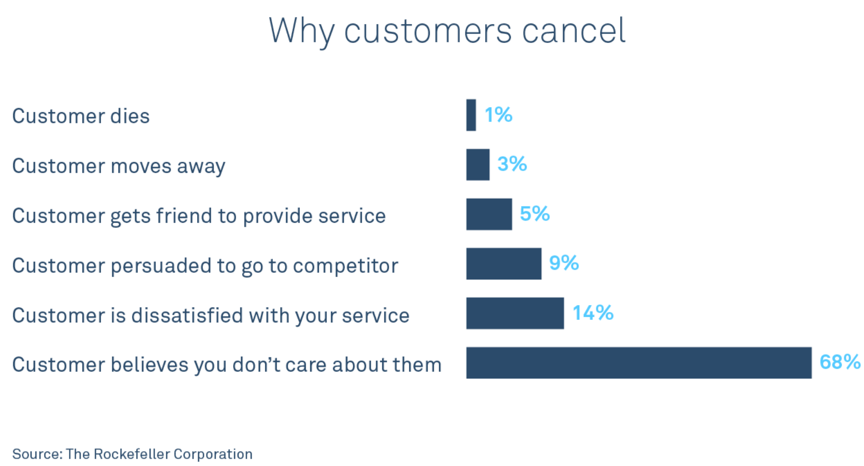
We should note finally that supporting free customers is a unique case (and we'll get into the pros and cons of that shortly). Many SaaS companies now account for supporting free customers as a sales and marketing expense – Box did so when it went public. The underlying principle is that those support costs aren't COGS because there is no sale taking place.

Rather than simply looking at the cost of support, it's worth considering how support can actually be a positive force on revenue. Here's just three ways you can position support to grow your business.

Support to convert

If you provide support to customers on free trials you need to think about what's a sales cost and what's actually customer support. If you are supporting those users to convert them to paying customers then it's a sales cost. If someone is on a trial of a $25,000 a month plan and they have a question three days into it is that actually COGS or is it a sales cost? It's sales because if the potential customer doesn't get what they want they are not going to buy the product. It's not COGS because you haven't actually sold a good yet.

Support to preserve



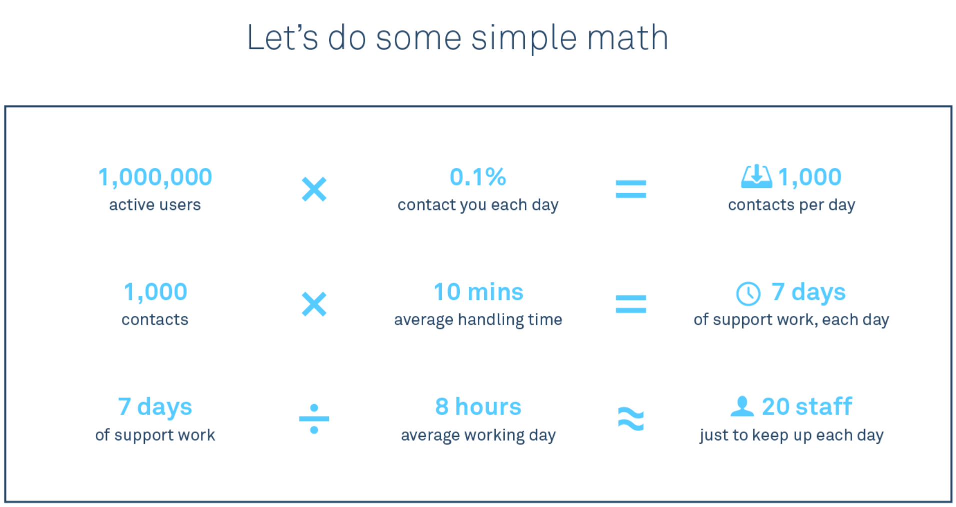
The number one reason customers quit is because they believe you no longer care about them. There are numerous studies that show this to be true, including the one above from the Rockefeller Corporation. Through that lens you can start to think of support as something that turns an 18-month relationship into a three-year relationship and doubles your revenues in the process. Basecamp have customers that have been with them for 12 years, since the product first launched. That's because they've built a good product, but also because they've consistently provided great support. Support can increase LTV if it preserves the customer relationship.

Support to distinguish

This is support as a clear differentiator that can be used in your sales and marketing efforts. A good way to think of this is as follows; most car companies see reliability as something they grit their teeth about. Toyota uses it as their number one characteristic. Similarly financial services company Discover ran a TV advertising campaign that focused on the fact that when you called its support line you would get to talk to a real person.

Hiring is not a silver bullet

You might think the solution is just to hire a huge customer support team, but that doesn't scale. If you have 1 million daily active users (DAU), 0.1% of them contact you each day, and each support conversation you have is on average 10 minutes, you'll need 20 full-time support employees just to keep up each day.



That's 20 support staff with absolutely no wiggle room for things like support requests that take longer to handle than average, an employee who is sick or wants to take vacation, the roll out of a new product or feature which generates lots of queries, etc. If any of those things happen you're going to start falling behind. The only way to start catching back up is to increase the workload or hire more agents. That doesn't scale, which is why hiring is not, and can't be, a sustainable support strategy.

There are plenty of other tempting ways to "fix" the problem of support as you grow. But none of them will benefit you in the medium term and your customers won't thank you for implementing them.

Triage out

This involves offering some kind of VIP support for high value customers, who get a simple and easy way to contact you. You then spend all of your time and effort making sure these VIP customers are properly supported. But the rest of your customers are treated like passengers on a crowded flight with a budget airline. You keep them at arm's length by getting them to fill out complicated forms and jump through multiple hoops, just so they can talk to a human. In effect you are forcing the overhead of providing support back on to your customers.

Busy out

This isn't so much a strategy as a total failure of your support organization. You get so overloaded and backlogged that you start ignoring your incoming contacts, or give them such poor support that customers give up on contacting you. Either way you are no longer faced with the challenge of scaling your support team.

Burn out

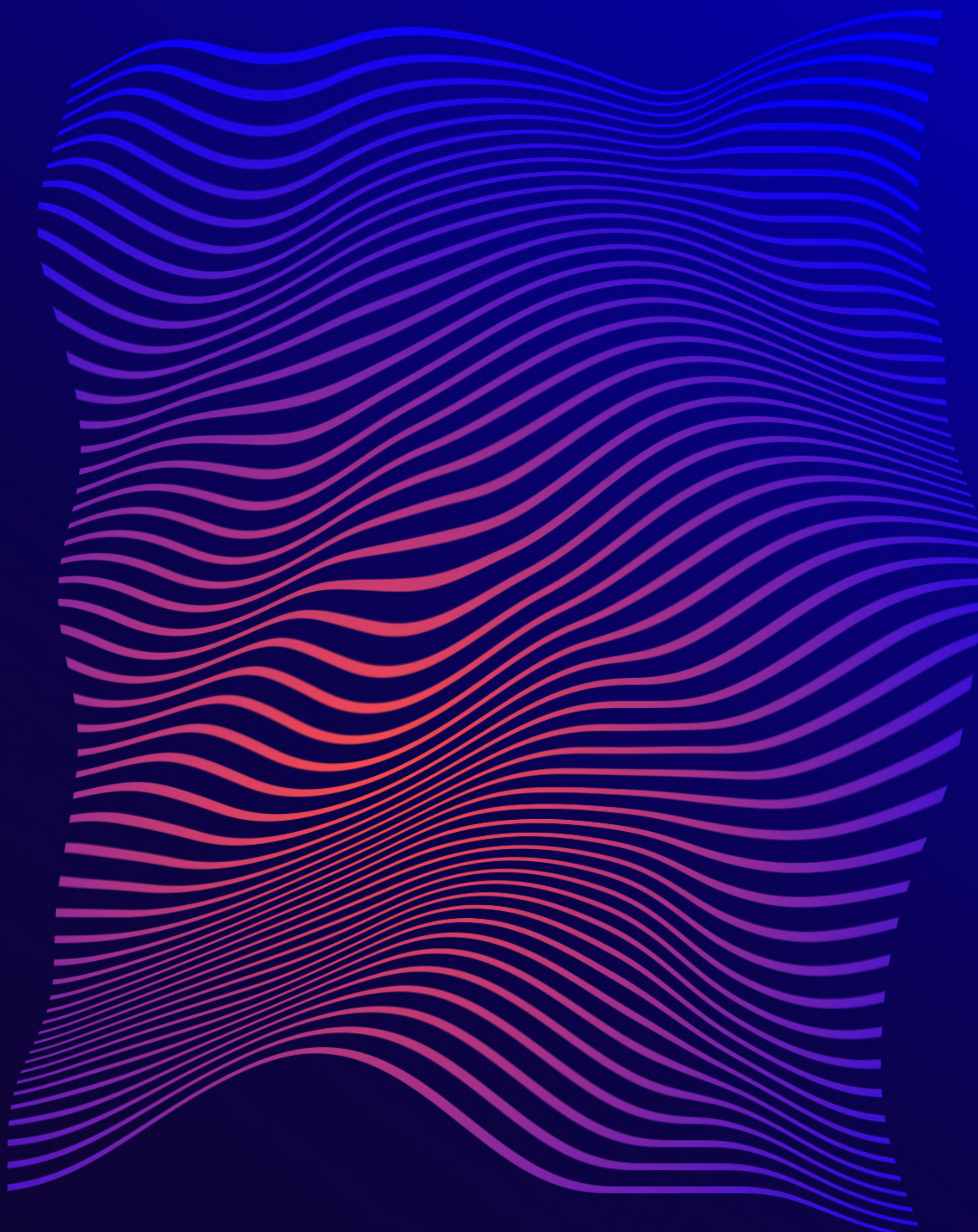
Burn out is what happens when you fail to give your support team the bandwidth and resources to do their job. It quickly leads to a high staff turnover rate, a perennial problem in customer support. Imagine if you ran a restaurant and every few months you got an entirely new kitchen staff but kept the menu the same. All the experience, deeper knowledge and understanding your kitchen staff had walks out the door and you start from zero again. That's what organizational amnesia looks like.

Outsourcing

The extreme form of outsourcing is hiring a third party, probably in a low cost economy on the other side of the world, to handle customer support. But hiring your own support team and then marginalizing them causes the exact same issues. Outsourcing ensures customers will get replies, but with no guarantee about the quality of those replies.

Building out your team

Chapter / 02

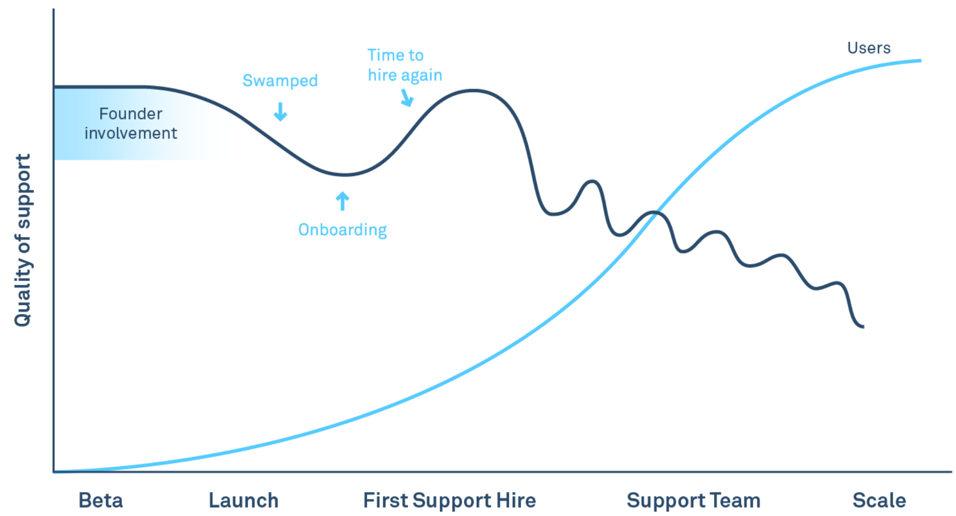


Hiring should be a core competency of every startup. In order to grow you need to hire the best people, and if you stop growing for any protracted period, then you're probably not a startup anymore. You're either a mature company or you're dead.

When it comes to building a world class support organization, hiring is arguably even more crucial. Your support team is the public face of your company and you are under a lot of pressure to get it right. Bad hires damage your company's reputation every time that person has a conversation with a customer. Just as importantly bad hires damage your reputation with your team.

Supply and demand

When considering your hiring needs, it's useful to view your support team through the lens of classic economics. Your job is to match an appropriate level of supply (new hires or efficiencies that allow each rep/engineer to accomplish more) to a given level of demand on their time (questions from your customers, writing and updating help content, tracking feedback, etc.).



Drivers of demand

Your next step is getting a handle on the amount of work your team faces on a daily, weekly, and seasonal basis. At Intercom, this is measured by the number of conversations each team member is having. If your support team is responsible for updating or writing docs, product research, or other tasks don’t forget to factor that into the workload.

In addition, knowing what time of the day and day of the week you’re busiest gives you extra depth. Unless your business is narrowly targeted to a single city or region, you are going to see the number of contacts viewed across any given 24-hour period probably has one or two spikes. For Intercom, these spikes coincide with the start of the working day in Europe and on the west coast of North America. Your supply (your support team) will have to match these spikes appropriately.

Supply considerations

Clearly understanding the type of work required means you will hire people with the appropriate skill set.

If your product requires new customers to crack open a code editor and push to production then your support team must be well equipped to guide them through the process. As a result it is going to be engineering heavy. On the opposite end of the spectrum, it is a complete waste to have an engineer answering basic, and possibly monotonous, questions about how to use features. In this case your team will skew more heavily towards support reps.

And while it might sound obvious don't forget holidays, sick days and other time off for your team. There is nothing worse than receiving an auto-responder from a business explaining that they are observing a public holiday in their home country. Similarly even the most motivated team is going to need to take unscheduled time off work due to things like illness or family emergencies. These are not problems you should be passing along to your customers so factor them into your model and plan accordingly.

Finding the right workload

Once you've established your levels of supply and demand and how you will track them, the missing link is calibrating what a busy – but manageable – workload looks like for an average member of your team. Be warned, it takes time to arrive at this figure and the only way to do so is with hard earned experience.

While Intercom's first support hire was made relatively early, we were much slower to hire our second, third, and fourth members of the support team. While keeping up with the workload was extremely challenging, it meant we found the line between high – but sustainable – output and overdoing it. Knowing what a sustainable workload looks like gives us an accurate picture of the number of people needed to meet different levels of demand. The hiring model we've built using that data enables us to always hire someone a few weeks before we begin sacrificing the quality of our responses due to overwork.

Hiring for support

Hiring is hard for any position, but when you are hiring for support, you're looking for someone who is going to represent your company on the frontline. They will be talking to your customers all day every day, and customers' interactions with these people have a massive impact on how they think and feel about the entire company. You want them to feel like you are helping them succeed, like you value their business, like it's worth continuing to pay money for your product.

The decision to add a new member to the support team is not one that should be taken lightly or done on the spur of the moment. While tempting, it certainly shouldn't be rushed by feeling the pressure of a growing inbox. Although nearly every company will tell you that "every hire should raise the bar", most companies can't or don't follow through. Especially when the backlog of customer issues is piling up and they desperately need more hands on deck.

This is why the previous sections of this book on the key metrics for support and how to model growth are so important. A metrics-driven model of your support organization's growth enables you to make informed predictions about when you'll need to make a new hire and get ahead of the curve. Use that breathing room wisely and you'll minimize the chances of making a poor hire.

Traits to hire for

It's one thing knowing when you need to hire. It's another thing entirely knowing who you should be hiring.

Pret, the high-end cafe chain, says you can't hire someone who can make a sandwich and teach them to be happy, but you can teach happy people to make a sandwich. Something similar applies in software support. The ideal support person has a blend of technical knowledge and customer empathy. Technical skills can be taught but teaching someone to operate in a way that isn't natural to them is nearly impossible.

That's why it's a good investment of your time to draw up meaningful values for your company as a whole but your customer support team in particular. Well thought out values, which everyone in your company has bought into, provide a benchmark against which you can assess job candidates and also assess performance once they've joined. When hiring for customer support, first and foremost you should look for people that display values in line with your own.

Probably the best way to illustrate the power of team values is by sharing two of our own and what they mean in practice.

Thrive under pressure: We remain calm, collected, and creative under immense pressure. We are resilient and know that our ability to remain unhindered by a customer's emotional response to a situation – be it screaming, belittling, appealing for "someone with authority", etc. – allows us to focus on the issue and potential solutions. We delight in remaining engaged until we have solved every last issue and answered every last question.

Treat the problem: We are always looking beyond a customer's question for root causes, always asking why (multiple times and in multiple ways if necessary), and we never settle for the status quo. We understand a customer's question is rooted in a goal they are trying to achieve and we are always on the lookout for non-traditional or non-obvious ways that they might achieve that goal. We act as the advocate and voice for the customer within Intercom with the goal of continually improving our product.

Secondly, we look for people that demonstrate a high degree of self-awareness, discipline, and self-control. People that remain calm during an emergency and can control their emotions when customers don't. We prioritize this trait of calm under fire based on our own experience of running a customer support org but also based on data from *The Effortless Experience*. Based on research with over 1,300 top performing customer support reps, they found the following were the most important traits reps displayed, in descending order of their impact on customer satisfaction.

1. CONTROL QUOTIENT

Displayed through:

- Ability to handle high-pressure situations without becoming burned out
- Taking responsibility for own actions
- Responding well to constructive criticism by managers
- Ability to concentrate on tasks over extended periods of time

2. EMOTIONAL INTELLIGENCE (EQ)

Understand customers better than anyone else. Customers are experts in their problem, but know that most of the time they don't have the best solution. Understand that a user's question is only their way of externalizing a problem they are trying to solve. Strive to always be helpful, kind, and ego-less. Displayed through:

- Empathy
- Ability to respond to different personality types
- Customer service ethic
- Comfortable interacting with strangers

- Advocating for the customer
- Persuasiveness

3. BASIC SKILLS AND BEHAVIORS

Including:

- Product knowledge
- Technological expertise
- Confident and effective communications
- Asking good questions
- Ability to multitask

4. ADVANCED PROBLEM SOLVING (IQ)

Able to structure and process qualitative or quantitative data and draw insightful conclusions from it. Exhibits a probing mind and achieves penetrating insights. There is always something new to discover, and there is always an opportunity to dig deeper on a topic you are already familiar with. Don’t wait for someone to teach you, find the answers yourself.

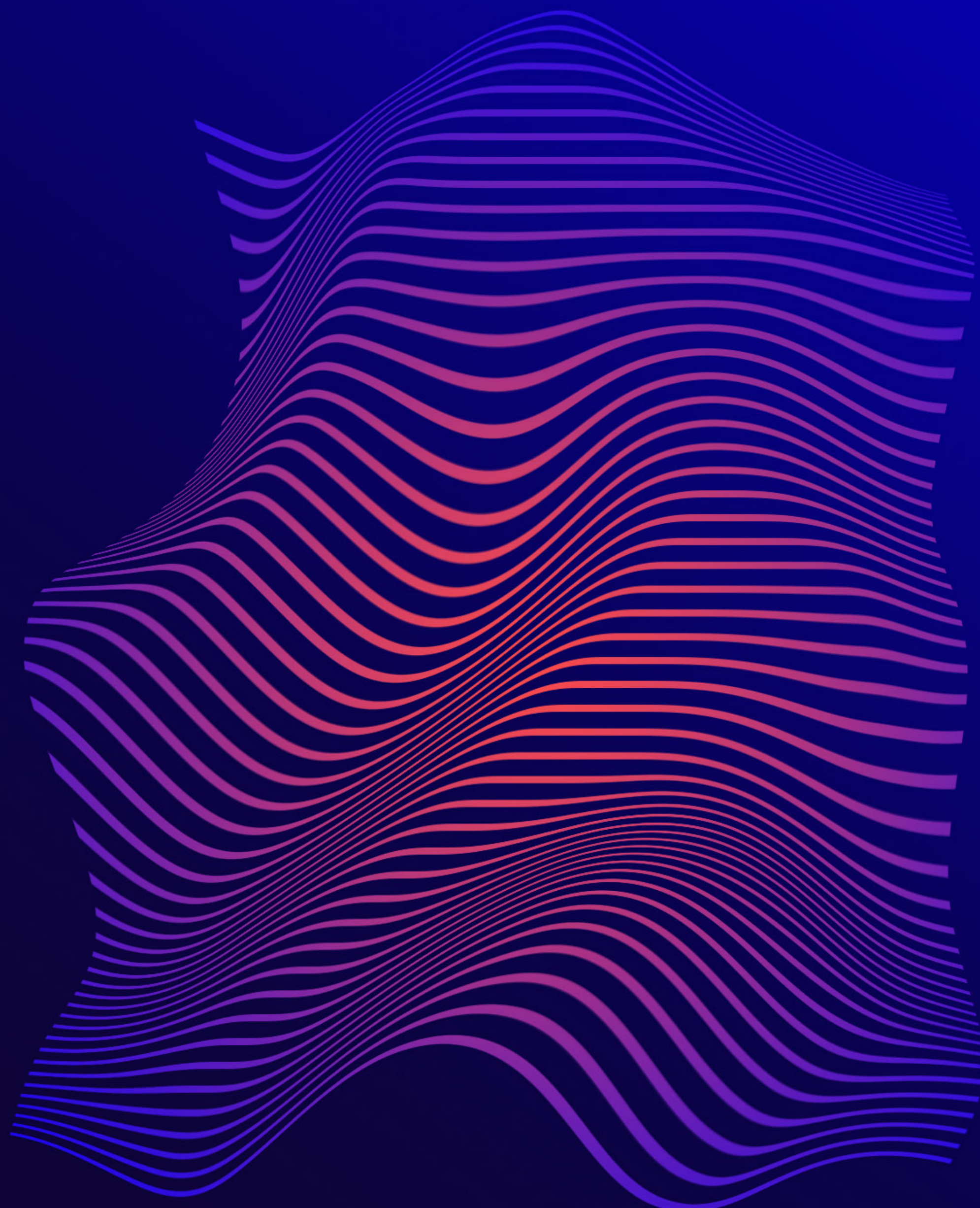
You are looking for an ability to quickly and proficiently absorb, understand, and use new information. Displayed through:

- Curiosity
- Creativity
- Critical thinking

- Experimentation

Choosing your level of support

Chapter / 03

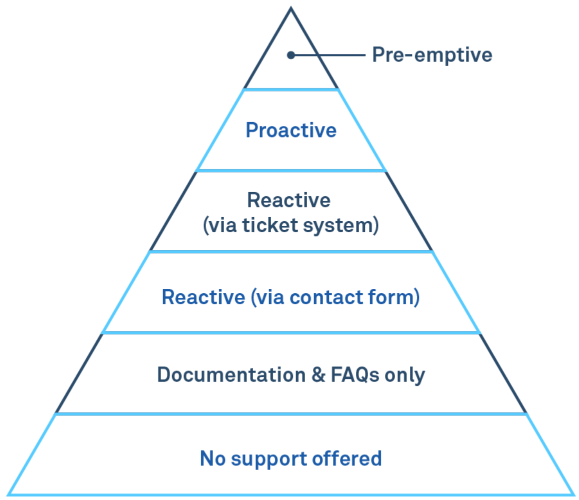


The very second someone asks you a question about your product you're in the business of supporting your customers. But what experience do they expect? What experience do you want them to have? What experience can you afford to provide? Your gut answer here will usually be something like "the best experience possible". But that begs further questions. Best for whom? What does "possible" actually mean here?

Every business assumes it should support all its customers in the best way possible. As they scale this will become prohibitively expensive. When the product is in its early stages, yes, it makes sense to have direct individual conversations with every user you get. But as you scale the ability to profit off the relationship with the customer, and not be burdened by the cost of overcommitting support to them, becomes essential. After all, we support customers so we can continue to profit from them, not to lose money on them.

The hierarchy of support

It's obvious to most people that a customer spending a million dollars a month and a customer spending a dollar a month will have different expectations of your support. Does this mean that they should receive a different support experience? We would argue probably. The manner in which you can afford to support a customer will be necessarily guided by the potential revenue to be extracted from a healthy, committed, long-term relationship. You also need to define the various levels of support you can offer and how should they be paired accordingly. To think about this consider the following hierarchy of support:



No support

At the bottom of the hierarchy is of course no support. Zero. You literally provide no material whatsoever to explain your product and its offerings to your users. This might sound insane but there are some consumer apps that provide little to nothing by way of explanations, tips, or tools to help anyone become users. And in many cases it hasn't impeded their growth at all. We're looking at you Snapchat.

Documentation/FAQs/forums

Most companies don't have the option of providing no support. The bare bones any consumer or B2B offering should provide is a layer of support through documentation and FAQs. It's self-service. There are no staff available to ensure the customer is having a good experience and is on the road to becoming an engaged user. You only put effort into the deflective nature of documentation – which is to ignore how powerful help content can really be. Many, many consumer products fall into this category, simply because the value of the relationship, the raw money that can be extracted from showing ads to eyeballs, is not high enough to justify any back and forth between the user and a support representative.

For example if a product monetizes by showing ads to a customer, and the customer sees on average one advert a day, maybe the total revenue for that user is $3 a month. If that user was to have even one reasonable conversation with support, that would nuke two years of their value to the business. That is why the numbers will never justify anything other than basic support through docs and FAQs.

Reactive support via contact form/email

Reactive support is the type of support that businesses most commonly provide. Typically it falls into two categories. First is reactive via contact form. This is when you encourage your user to submit an issue through some form that is multi-purpose on your website i.e. complaints and issues will be lumped in with job applications, PR opportunities, company enquiries and many other types of communication. The user is offered no guarantee that their complaint has been received, understood or will be acted upon in the future.

Reactive support via ticketing system

Ticketing systems typically have the following characteristics: they ensure that a customer knows where they stand in the pipeline; they normally set some sort of minimum expectation of response time, usually

24-72 hours; and they are designed to make sure that every user gets a response, even if that is a simple, blunt “no, thanks”.

Proactive support

Proactive support is often related to the new function of customer success being established in many companies. Proactive support seeks to identify problems customers are experiencing before the customer gets the chance to register their concern. For example, it might mean reaching out to a customer who has failed to upload a video three times and explain better ways to do it. Or it will identify someone who is sharing a file with zero teammates and explain how to add teammates. Proactive support is expensive and only justifiable when the resultant happy customer will continue to have a long and valuable relationship with your company.

Pre-emptive support

The highest level of support is no support necessary. This is when every problem is pre-empted through product design. Later we’ll discuss how your support team can work effectively with product to provide a positive feedback loop.

Match costs with the value of the customer

On a scale of no support to proactive support, the most important consideration is to pair the level of support with the value of the customer. Make sure you are never treating a valuable customer badly nor are you wasting precious support hours on a very, very low revenue customer. Ron Baker once wrote that bad customers will drive out good customers. This is the simplest statement of how that plays out; a good, valuable customer will have high expectations and if their level of service is dragged down by the burden of less valuable customers, you will end up with a discordance between what the customer thinks they are paying versus what you think you should be giving them.

As you can guess, it's for this reason that many B2B products offer premium support as part of their high-end offerings. It is where phone service, personal support agents and account management kick in.

The following table outlines the various tactics and techniques to deliver each category of service while defending your business from spending money where it doesn't need to be spent.

Don't assume all your users are equal

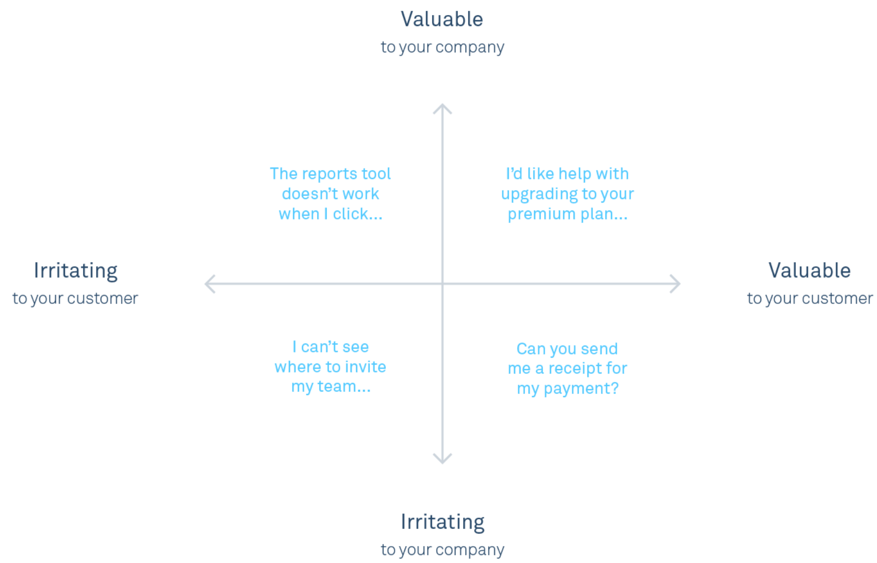
Level of support	Good for...	Bad for...	Challenges
Documentation	• Self service • 24/7 availability • Maintaining a knowledge base	• User specific issues • New issues	• Keeping docs up to date • Ensuring they are well written
Reactive (contact form)[1]	• Problem mitigation • Giving users a way to reach out • Letting companies triage issues	• Customer experience	• Setting expectations
Reactive (ticket system)[2]	• Triaging issues • Setting expectations • Letting user prioritize issues	• Personal service • Customer experience • Urgency	• Providing personal support • Customer feels like a ticket
Live chat	• Urgent issues • Providing personal support	• Workload • Letting company triage issues	• Staffing • Response times • Managing customer expectations
Proactive	• Personal support • Reaching more than just customers who complain • Success-driven support	• Workload can be significant • Costs	• Solving the equation to make it valuable • Integrating with other support tactics

1 - For many startups this may take the form of email to the founder

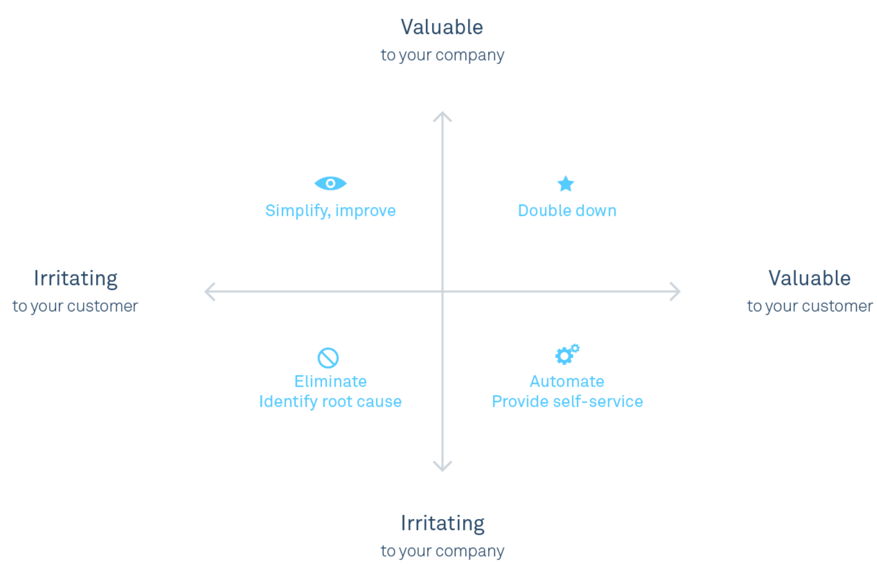
2 - By ticket we mean any systematic form of response that allows the customers to see their progress towards getting a response

Reducing the cost of support

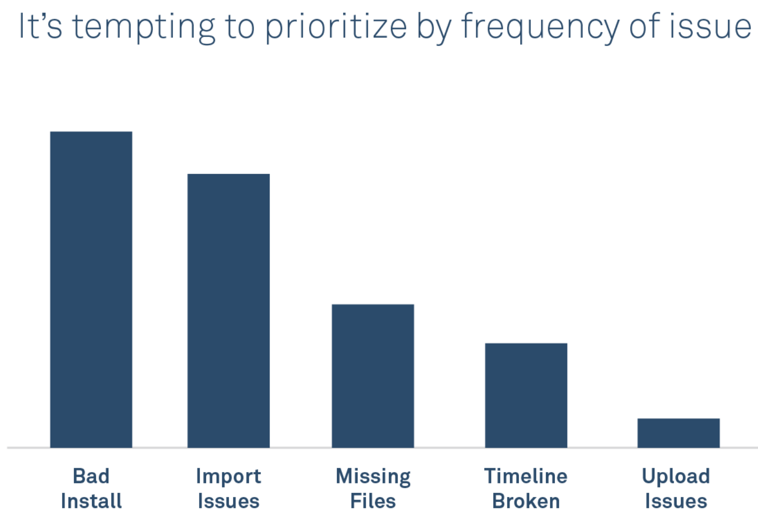
One of the key ways you can reduce your volume of support requests is to reduce the number of dumb contacts your team is dealing with. Every contact is either irritating or valuable, for your company and your customer alike. Dumb contacts are the ones that are irritating for both.



For each type of contact, there's an appropriate action you should follow...



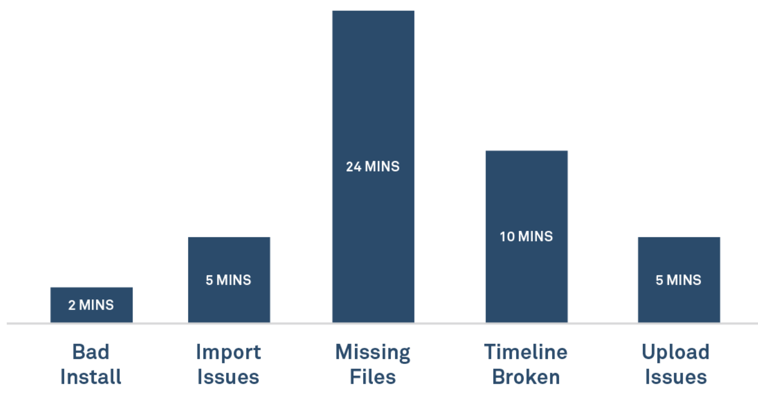
The next step is simple; quantify each type of dumb contact and how long it takes to resolve. This lets you prioritize.



The knee-jerk reaction here is to solve the most frequently occurring issues first. This is often the best way to tackle your volume initially, as it gets rid of the low-hanging fruit.

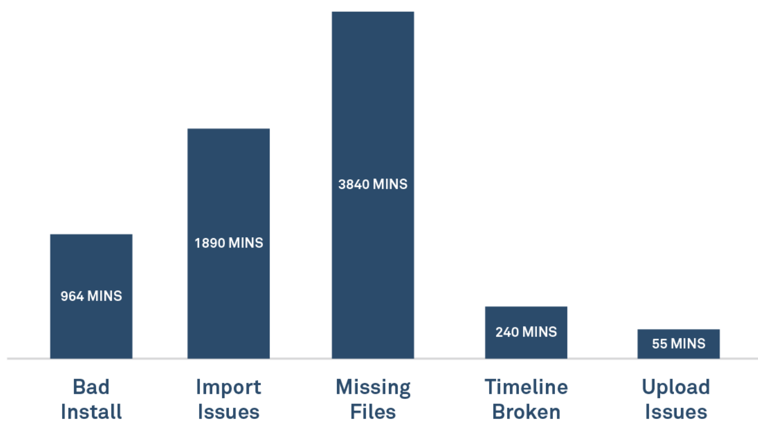
You can’t stop there though. You need to understand which issues take most time. This is usually called average handle time, AHT for short.

Understand the time to resolve each issue

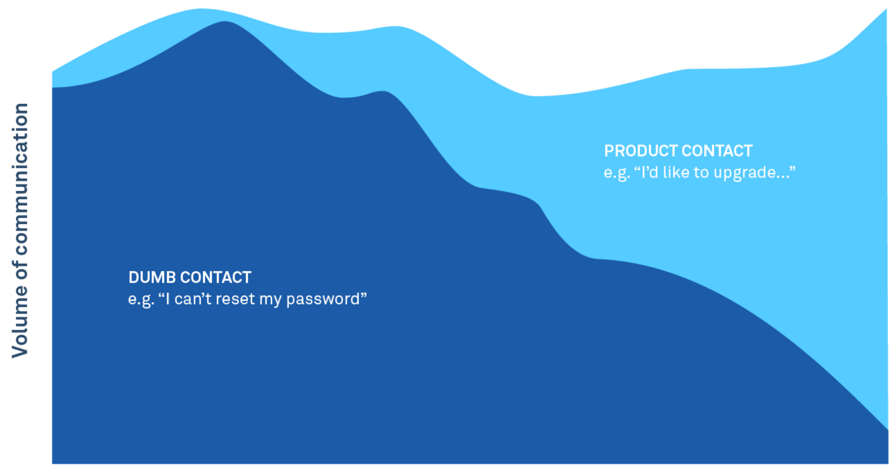


Combining the volume with AHT gives you a sense of where the biggest opportunity is.

Calculate where your team spends its time



Every category then needs to be thoroughly understood so it can be tackled appropriately. Solving the problem can involve a redesign, better copy, faster performances, more robust error handling, broader file support, or plain old bug fixing.



As dumb contacts decrease, you have bought yourself capacity to have meaningful communications with your customers. Ones where you help them get the most from your product, and ones where you learn lots about how it is used, and how it could be improved. These are the communications that grow loyalty and create evangelists.

Should you support free customers?

When deciding whether or not to support "free" customers, the first question you have to ask yourself is, what do you consider a free user to be? There are many kinds of people who use your product but pay you zero dollars.

- **Customers on free trials.** They are still evaluating your product.

- **Short-term free customers.** They didn't convert straight after the free trial but are still actively using and engaged with your product.
- **Long-term free customers.** They are long time users of your product who have never spent a single dollar with you.
- **Lapsed users.** They paid for your product but no longer do.

Related to this question is how you generate revenue. The preceding examples assume revenue only comes in the form of a payment from the end user. There is of course a second type of revenue - where someone else pays to display things to the user every month. In those cases - typically B2C apps - there is no such thing as a free user, just as there is no such thing as a paying user. There are just users. No single individual's eyeballs are worth more than anyone else's.

Free trials

When a customer is on trial you have to assume the best case scenario - that this is the start of a long, fruitful and profitable relationship. You have done the hard work of attracting them to your website and convincing them to sign up. They are now signed up and are engaged in a trial. You have to believe in the best possible outcome and treat them as the best possible customer. So when someone is on trial it is imperative that they have an excellent experience. It is not just your product that is on trial, it is your company. It is your sales, your marketing, your billing, your administration and of course, your support. Yes, the major factor that will make or break the deal is the quality of your product. But a lousy or late response from support during a trial will end the relationship just as quickly. Therefore, if you are offering a free trial, you must treat the people who are on the trial like you treat your best and brightest customers. For all you know one day they will be your best and brightest.

Short-term free

Short-term free customers are typically folks who have emerged from trial but not been convinced your product is worth the money. They are in a sense still evaluating you, albeit now with restrictions in place. If you believe your free plan is sufficiently restricted so that engaged, satisfied users will be led to upgrade, then you should continue to treat them just as well as your long-term paying customers. Note the phrase “will be led to”. A well-designed free trial will force an upgrade when the user hits a certain limit in terms of usage, adoption, engagement, storage, etc. An upgrade should be the natural conclusion of a happy and productive engagement. Whatever continued usage looks like it has to lead to upgrading. Flashing a banner telling users that they are over their plan’s quota is unlikely to do the trick if they still have access to the features they want. But if, for example, you limit the number of messages a user can send to five, then continued usage is going to lead to upgrade.

If your free plan is such that it doesn’t automatically trigger an upgrade after a certain period of usage or time, these users will turn into our next category.

Long-term free

These are people who have been using your product long enough to make an informed choice not to pay you for it, and there is no credible reason to believe they ever will. They are satisfied with the offering you are giving them, continue to use and enjoy it, but have chosen not to upgrade. It is tempting to believe that one day in the future they will be a company big enough or will somehow find the revenue to upgrade. This is a pipedream. Your business should not be dependent on the future success of other businesses. You are not a venture capitalist.

This begs a question. Why do you support people who have never paid you a dollar? The only reason you would do that is if they are paying you in some other way than direct revenue. Below are some examples of that and also the kind of hard metrics you will want to measure to see if it makes financial sense to support these customers.

Positive word of mouth from a free user is advertising that you are buying and you are paying for it with the costs of your support, hosting and infrastructure. Is it worth it? You have to do the math. The last piece is crucial. If you are saying providing a free plan and supporting people on it is a marketing activity, then you need to cost it as a marketing expense and measure the return on investment.

Form of payment	Example	How to measure
Word of mouth	Tweets/shares	Number of tweets that lead to new revenue
By night/by day	Using a product on a side project	Number of users who have both a free and a paid plan
Product exposure	Embedded badge to promote product to a customer's site	Traffic from the badge that leads to revenue
Network effect	Invites to new users	Signups from the imported networks
Eyeballs	Banner advertising	Revenue from ads

A customer paying you zero dollars who has a question that takes 30 minutes for a support rep to deal with, costs you 30 minutes of the rep's time plus the customer's allocation of your fixed costs i.e. hosting and infrastructure. Typically it's near impossible to split or vary their impact on your hosting and infrastructural cost base, as it's needlessly complex - they are using the same app as your paid customers. But the customer support piece is easier to control. You can make the decision of whether to support them or not. Secondly, you can decide how to support them. To recoup a 30-minute conversation might require a lot of product exposure or word of mouth. But that's not to say the equation will never make sense. It is just entirely dependent on your business model.

But you do have the ability to limit the cost implications for your customer support team.

1. You can de-prioritize them.
2. You can move them up or down your hierarchy of support accordingly, e.g. you may decide to only offer them self-service

support.

Lapsed users

There are two things you know for sure about lapsed customers:

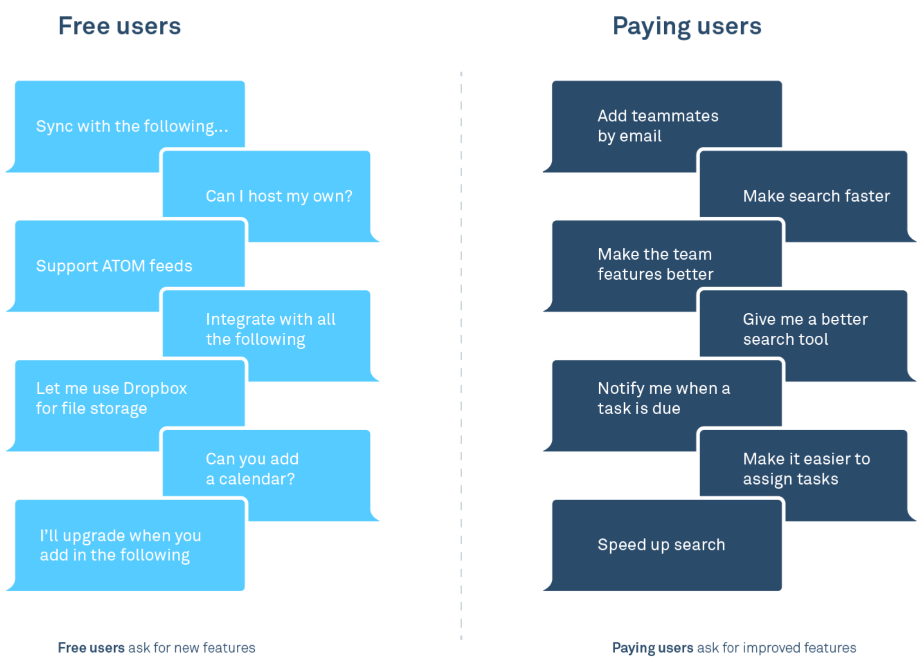
1. They definitely can pay you revenue
2. They have chosen not to pay you revenue.

So you have to treat them like a new customer - give them the best service and see what happens.

You should support your free customers if you can solve an equation where the ongoing support costs for free users exclusively is ROI positive in terms of:

- the amount of new customers generated.
- new revenue generated through things like advertising.
- your product is in its embryonic stages so word of mouth, awareness and publicity is disproportionately valuable as you are still trying to establish a brand.

It is tempting but often wrong to assume that product feedback from free customers will be invaluable to you. For sure, when you have no customers all feedback is welcome. But that changes once you have paying customers, paying you real cash out of their bank account to solve a real business or consumer problem. Feedback from paying customers is a far stronger and cleaner signal of where your product should go than what you will hear from people who have yet to give you any actual money. Free users always want more, paying customers always want better. More isn't better. Better is better.



The case against supporting free is simple. The majority of customers who do support their free customers are doing so without a full understanding of the costs of doing it. Previously we talked about the concept of bad customers pushing out good customers. This is a perfect example of where supporting free customers will lead you astray. All else being equal if you have 10 customers to support and 4 of them are free, the level of service you are offering the 6 who pay will reduce. Choosing to support free adds a significant workload, because in almost all cases free users will significantly outnumber paying customers. Most businesses are not savvy enough with their financials, marketing expenses and ROI calculations to work out whether or not it makes sense.

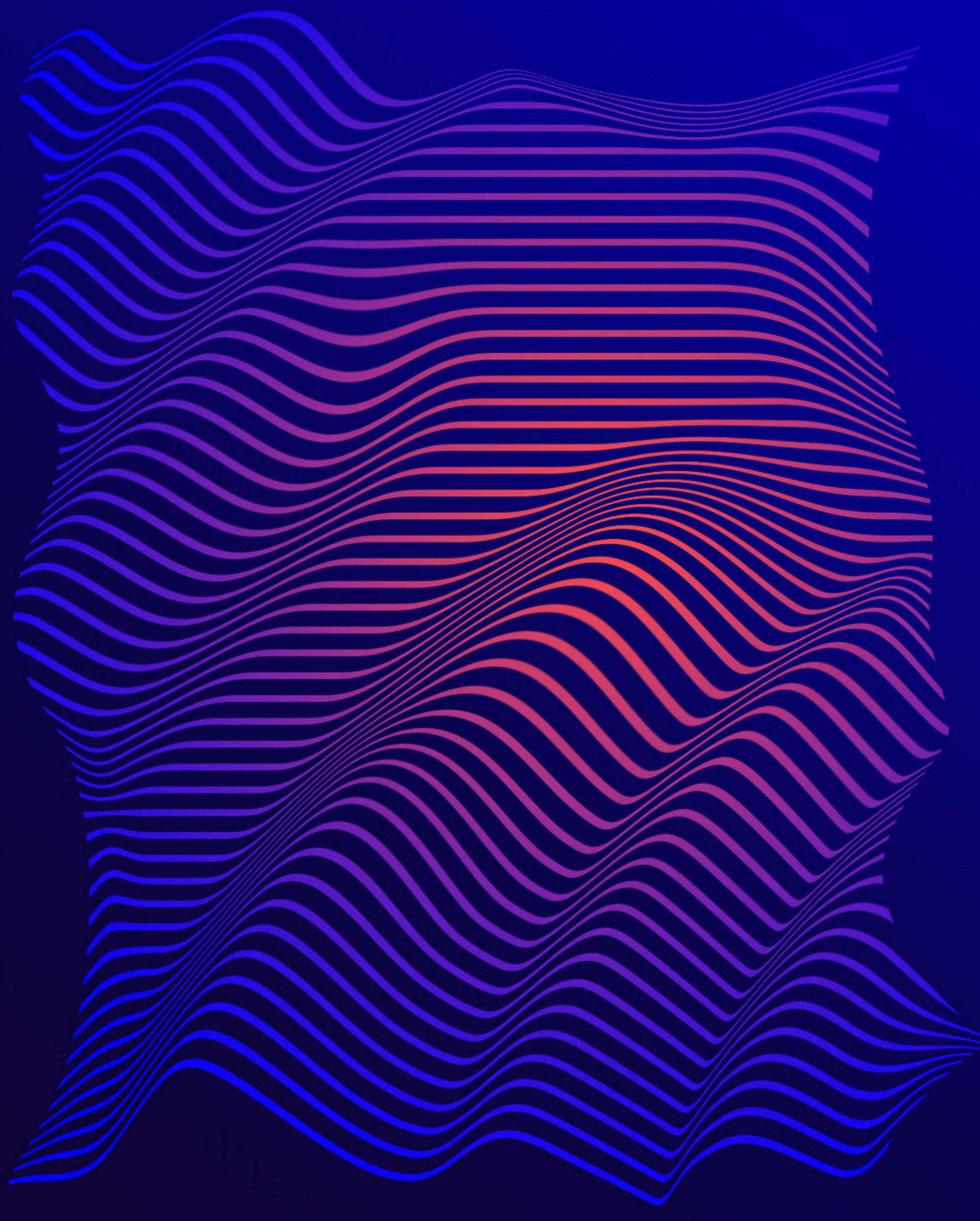
If it's not a no-brainer, it's a no-brainer. If it's not obvious you are getting something from your free users, chances are you are not getting anything from your free users. Adii Pienaar, co-founder of WooCommerce, echoed this particular sentiment in his homage to no longer supporting free, saying, "If they are not paying you, they are not your customer."

The question of if, how and why to support free users is multi-faceted. There is no simple prescriptive answer. It varies depending on how you monetize; how you define free; what activities you allow under your free plan; how you run your business - are you a side project, are you venture-backed, are you bootstrapped. And lastly the stage your business is at - embryonic or established. There are no right answers but this framework will enable you to make your own decision.

Defining your support

Chapter / 04

Intercom on Customer Support
—2nd edition

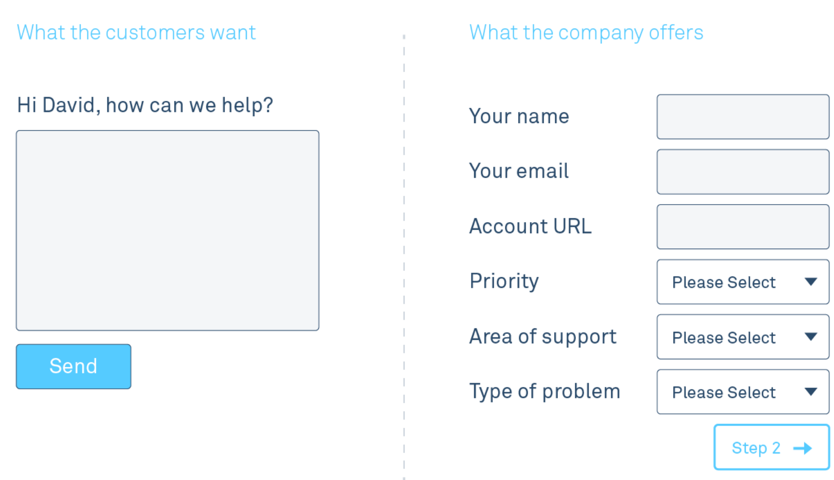


How do you expect to be treated when you deal with a government agency? What about at a Michelin-starred restaurant where you made a reservation months in advance? When your flight with a low-cost airline is delayed or cancelled?

Before a customer ever gets in touch with your team they have already formed expectations about your company, your support team, and your product. These expectations can be positive or negative, emotional or logical, subtle or bluntly-stated. The job of your customer support team is twofold:

1. To influence expectations before the customer ever reaches out for support
2. To help mold these expectations while the customer is waiting for your reply

Before the customer gets in touch



It is a mistake to focus solely on reacting to users once they are in touch. You can - and should - use all the tools in your arsenal to shape customers' assumptions about the quality and level of support you provide. Is it immediately obvious where I need to go for support? Can I find up to date and useful help content if I don't want to or can't get in touch with a human? How easy is it to get in touch with a human if I want? How long does it take to get a response? Do I receive a quick acknowledgment that I have a problem and need help? These are all things you can influence and make decisions about. Your choices have a direct bearing on customer expectations.

Using a knowledge base, documentation or help center solely as a way to inhibit users getting in touch sends the message that you don't want to be bothered. Is there anything more infuriating than a recorded voice telling you support is also available through the company's website? Similarly, sending out an automated email with a standard list of working hours once the user does find a way to get in touch does little to soften the blow.

If you want the support experience to start on a good footing you should always make it easy and obvious for a customer to get in touch. Give them an accurate idea of how quickly you normally respond on that day of the week and at that hour of the day. This allows the customer to plan their next move based on real information. If your team normally responds in 15 minutes, the customer might choose to wait on your response in order to complete their task. However, if you get back to customers only after a day or more then the customer knows they can move on to doing something else.

You'll hear many arguments against sharing this kind of information with your customers - even from within your own organization. Most of them come down to just one thing. Teams are afraid of putting this information in the hands of customers in case they use it against them; "You answer most queries in 15 minutes at this time of day. Why did it take an hour to get back to me?"

Be better than that. Support shouldn't be adversarial, so don't get off on the wrong foot by acting defensively. Trust your customers to be reasonable human beings, which most are, and you'll be repaid in kind.

After the customer gets in touch

After a customer has contacted you their expectations continue to evolve. You need to frequently and accurately assess those expectations and ensure they are in line with what you can provide. You should have a consistent process to ensure timely responses. This can be as simple as first come, first served or as complex as various levels of service-level agreements (SLAs) and other contractual obligations. What's important is that you strive to respond in a way that is consistent with the customer's expectations. If you prioritize paying customers then the customer on a $500 a month enterprise plan should have different expectations about your level of responsiveness than someone who is using your ad-supported product. If they don't you've got a problem.

Many software companies operate some kind of VIP support level. This can be an explicit SLA on certain custom deals or price plans, or it can just be an internal emphasis on making sure your most valuable customers have a great experience with a little white glove treatment. Your VIP group could be made of up of:

1. Customers on your highest cost plans
2. Big name customers (everyone from influential bloggers to employees at big brands who may be using your product on a side project)
3. Well funded startups (databases like Crunchbase are useful for getting this data)

Whether or not you chose to tell customers they are now part of your VIP program is entirely up to you. But remember if you do, it's essential that the expectation you set of your VIP support matches what you can deliver.

Mismatched expectations

In the vast majority of cases, customers just need an answer to their question. Your job is to make sure they can get that as efficiently as possible in the manner they prefer, whether that's through help content, an in-app conversation or on the phone. If it is delivered in a timely fashion, then the interaction ends effortlessly. If there is a mismatch between what you can offer and what a customer is expecting, you need to take control of the situation. Tell the customer what you can do, when you're going to do it, and how things will progress from that point forward. Don't shy away from letting a customer know that their request isn't reasonable or isn't something you can commit to building.

Moments of delight

You should strive to create occasional moments of delight for your customers through your support. By definition, a surprise arises when your expectations don't match an outcome that you've just experienced. The difference in expectations doesn't need to be large.

You and your team should be constantly on the lookout for quick, lightweight ways to delight your customers (but don't let delight become your default strategy, as we'll explain shortly). Take an extra minute to find out what your customer's business is and tailor your answers to their specific situation. Send handwritten notes to new signups. Order a cake for a customer badly affected by a bug. These are relatively cheap and simple ways to create a lasting impression that you care beyond just the monthly recurring revenue a customer generates for your business. They also become fun moments for your team; unexpectedly delighting someone is just as beneficial for the giver as it is for the receiver.

Research and test your processes

Setting expectations clearly - even before a customer has a question - is a valuable weapon in your arsenal and one that doesn't require extra people, hours, or money. Take the time to test your support flow from end to end, with real customers that have real questions. Just as a finely tuned product team researches and tests new features, you should be researching and testing your process with customers and optimizing for clarity. Otherwise how can you ensure customers will get what they expect the next time they have an issue?

What is great service?

Defining what level of support you will offer is fundamental to everything that follows. But before talking about defining your support we'd like to counterpoint a notion that's gained a lot of currency in recent times; that great service is synonymous with over the top efforts to delight and wow your customers.

A lot of startups mistakenly think the path to customer support success is through surprising, delighting, and creating memorable experiences for your customers. After all, startups are all about doing things that don't scale, right? While this is one path, there are several reasons you shouldn't go down it by default.

The best known practitioner of the strategy of going above and beyond to delight customers is Zappos.com. A popular example of its gold standard customer service is the 10-hour phone call one of its customer service reps logged.

What is overlooked is that CEO Tony Hsieh systematically built the entire Zappos.com culture around going "above and beyond". Most businesses, and particularly startups, can't muster the resources, time, or stamina to do the same. And as with most things related to support, doing it half-assed is more damaging than doing nothing at all.

In the vast majority of cases customers don't want red carpets, Dom Perignon, and caviar; they just want you to fix their issue and get out of the way so they can get on with the job they were trying to accomplish.

The only case in which delightful over-the-top service makes sense is if it's baked into the absolute core of what a company stands for, a la Zappos. "At Zappos.com, we decided a long time ago that we didn't want our brand to be just about shoes, or clothing, or even online retailing," explains Hsieh, "We decided we wanted to build our brand to be about the very best customer service and the very best customer experience."

Aim for low effort, not delight

It turns out low effort rather than exaggerated shows of affection are the real driver of customer loyalty for most businesses. Shortening the path between asking a question and getting an answer lowers the objective effort required by customers. But it's even more important to lower the subjective, or perceived, effort. According to research cited in *The Effortless Experience*, "customer effort is actually mostly about how customers feel. The exertion required from the customer makes up only 35 percent of how they evaluate customer effort. But the interpretation side – the softer, more subjective elements based entirely on human emotions and reactions – make up a shocking 65 percent of the total impact."

The most important factor in customers perceiving low effort support is anticipation. The ability of your customer support team to get in front of customer questions and responses. At best, you offer solutions before the customer knows that they have a problem. At minimum, you ask the right questions and ensure the issue is well documented and clearly defined.

The customer feels that they have been fully understood and your team has an accurate definition of the problem and a clear path to a successful resolution.

Measuring next contact avoidance

Large customer service operations focus heavily on “first contact resolution” (FCR), the idea that the first rep contacted should fully solve the issue. While FCR is a great goal, “next contact avoidance” is a more subtle and far more powerful concept you should be prioritizing. An example: a customer reaches out about your reporting feature. FCR dictates that you make sure the customer knows where and how to generate reports in your product. Congrats, you have solved their issue on the first contact.

Next contact avoidance takes things a step further by not only answering the obvious question, how to generate reports, but understands the logical next step for people generating reports is exporting them and so sends a link to documentation on how to do that as well.

Not only have you solved the customer’s current issue but also set them up for success in the entire job they were looking to do. Deep understanding of the jobs customers are trying to do enables you to predict their next steps and offer help that they don’t even know they need yet.

Defining your support

Everything that happens when a customer talks to support is an aggregation of marginal decisions you've made. Hopefully you've made active conscious decisions about what kind of support you are going to offer. And those decisions are the execution of the values that you've arrived at earlier on in your support team's evolution. But even if you haven't consciously arrived at decisions, that is a choice that will have consequences for the kind of support your customers receive e.g. if you don't provide support out of office hours, then you've designed it that you don't support people during the weekend even if they are paying you $3,000 a month.

Experience of customer	Decisions you did or didn't make
During your regular hours of operation, they wait six hours for a response to a question	You decided speed of responses was not a priority
Based in Indonesia, they submit a question and hear nothing for 16 hours	You chose not to have support in that time zone
Incorrect information is provided, which fails to fix the problem	You chose not to invest adequately in your help content so out-of-date advice is still being provided to customers
Despite spending $2,000 a month, no one reaches out to them when they tell a support rep they are about to quit	You decided not to have an escalation policy in place for when VIP customers are about to quit

None of this is an accident and you should never see your support as something that happened to you e.g. "You'll never guess what *happened* when Jenny contacted support with a billing issue?" That's not how it happens. Instead you should be thinking of it like this: "You won't believe what we *designed* to happen when Jenny contacted support with a billing issue."

As with all assumptions, you will need to revisit these on a regular basis. Responding to all queries within 30 minutes may make sense when you have 1,000 paying customers, and a corresponding level of queries, but can you scale that to 10,000 customers and all the additional support requests that come with that growth? What works for you now could easily be buckling under pressure in 6, 12 or 18 months time.

Here are some of the key features of your support that you get to design and that you should address as early as possible.

Style

A fundamental question you have to ask yourself is what style of support are you going to provide. One option is what you might call the "one big answer" approach. This involves trying to answer each customer contact with a comprehensive reply that covers every possible related scenario e.g. your answer may be five or six paragraphs long, include links to your documentation, and even have an embedded video about how to use the feature in question. It's comprehensive but there's no sense you want to engage in any back and forth communication with the customer.

The alternative is to have a conversation with your customer - whether that's via email, through in-app communications or over the phone. You try to get to the root of the issue and if appropriate you follow up with additional questions or advice.

Voice and tone

Closely related to the style of support you plan to offer is the manner in which you plan to speak to customers. You need to think about your company voice (e.g. formal and reserved or relaxed and chatty?) as well as the tone for different scenarios (e.g. responding to a customer who's been overcharged compared to speaking to a customer on Twitter). Some of the questions you might want to ask yourself include:

- How formal do you want your customer communications to be?
- Are you going to adopt a conversational tone?

While it's important to have these guidelines in place you don't want to be too prescriptive either. The last thing you want is to create a team of robotic agents running through a script of predetermined messages, never helping or sympathizing, just programmatically following instructions.

Quality

It may seem quite subjective but you also make choices about the quality of your support. Think of it this way: you can decide not to put your best engineers on support, which means your engineering support is not as good as it could be. But given the complexity of your product maybe your best engineers are not needed to work support. Maybe it's better to hire less experienced engineers but empower them to work closely with your product engineers when and where necessary.

Who you hire is crucial to the quality of support you can offer. Quite often this becomes a trade-off between technical ability and a candidate's ability to empathize with customers while staying calm under pressure (see our earlier chapter on hiring). A word of caution: once somebody has demonstrated an ability to learn programming it is relatively straightforward to improve their technical knowledge. But it's hard if not impossible to train someone to be resilient, empathetic, control situations, and to handle stress over long periods of time while continuing to be positive.

Speed

In an ideal world all customers would have a real-time conversation with a friendly and knowledgeable support rep every time they had an issue. But reality bites and we all know that's not realistic. So ask yourself just how important is speed of response?

You need to look at it from the point of view of your customers - how critical is speed of first response versus time to actually resolve the issue? You may also need to consider what competitors or challengers in your space might be providing. The good news is that speed, in common with coverage and language support, is largely a money problem. Hire more staff and you'll be able to answer your customers' questions quicker, in more languages and at all hours of the day and night. Whatever you decide, ensure you are tracking your average response times and time to first response to see if you are hitting your goals.

Coverage

Are you going to provide support 24 hours a day, 7 days a week? Or do you think Monday to Friday, from 9am to 5pm will suffice? Just remember, even the most business-focused enterprise software products get usage out of office hours. What holidays are you going to observe? And will you provide skeleton cover during holiday periods or none at all?

Language

What languages to support and when to start supporting them can be a tricky decision. Whatever your views on it, the fact of the matter is that English is the *lingua franca* of the global software business. If you are in the B2B space you can probably go a long way only supporting customers in English; as of April 2016, Intercom had more than 13,000 paying customers in over 90 countries and only provides English language support. If your customers are global consumers you will probably need to think about supporting them in their native language at a much earlier stage.

Another important consideration is your international marketing. There are many services you can you use to translate your website into different languages, but you have to ask what the implications on support will be if you do. For example, what's the point in attracting customers who only speak Taiwanese if you don't have Taiwanese-speaking staff to support them? It's not a no-brainer – you are attracting customers that you cannot speak to. The implications of decisions like this are also an important reminder of why support needs to have a representative at the top table in your organization.

Process

If you don't have robust processes in place things will break as you scale and your quality of customer support will suffer as a result. While your processes will need to change and evolve as you grow, it's much easier to put them in place early than try and graft processes onto work practices which have developed organically and are ingrained in your team's culture.

As the team expands process makes it easier for everyone to do a great job. No one is left wondering what they need to do - it's clear what is required in a number of defined situations. Outside of those scenarios you also want to make sure team members still feel empowered to make the decisions that are needed.

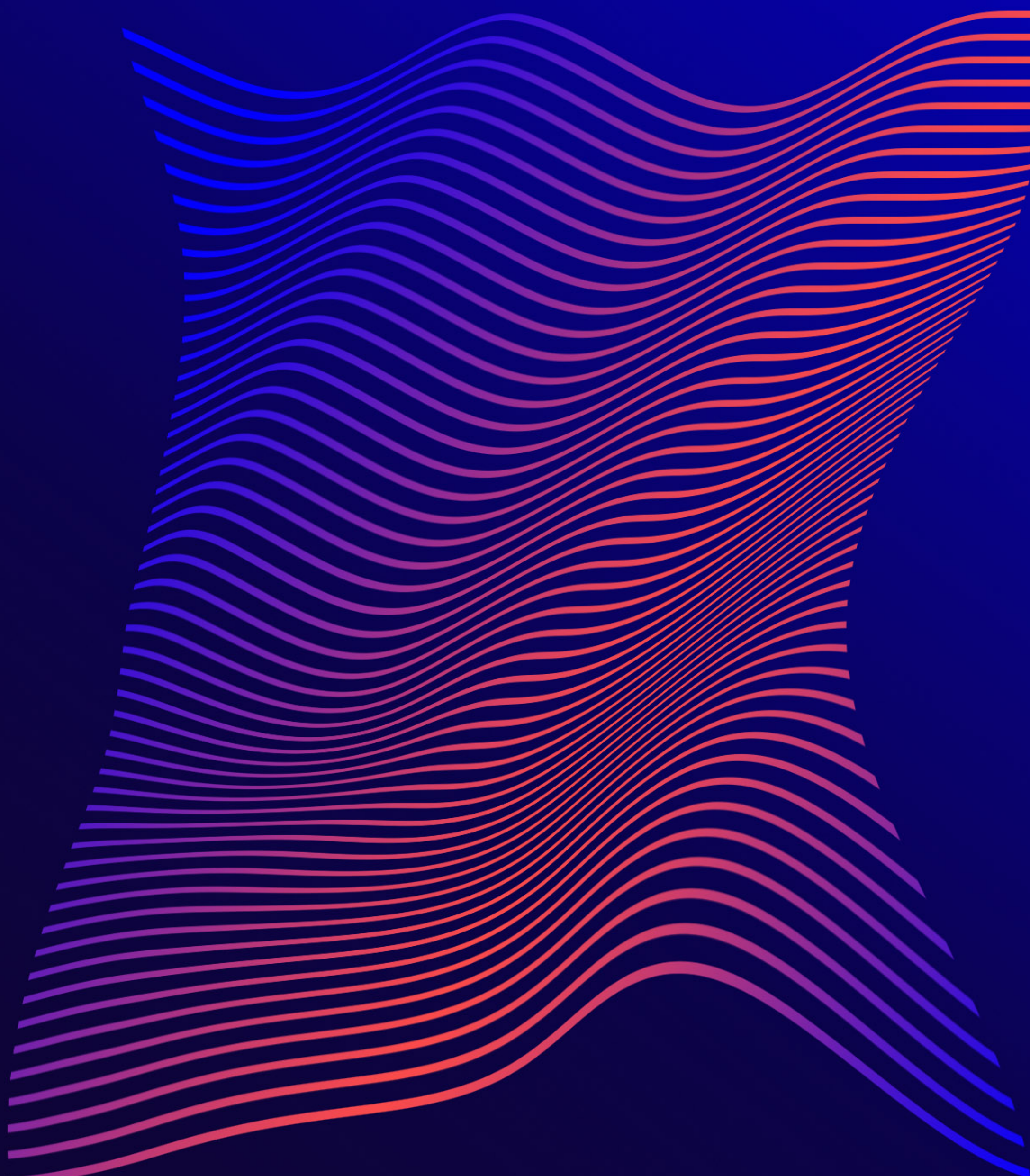
At the very least you want to make sure you have processes in place around:

- **emergencies:** how do you define an emergency and who gets notified? How are they informed and when?
- **escalation:** needs to be defined not just for emergencies e.g. for product bugs, when do you need to pull in a product engineer?
- **communication:** how do members of the team find out about stuff?
- **refunds:** under what circumstances will you issue them and who processes them?
- **security:** e.g. if someone asks to reset their password how do you

verify their identity?

Empowering your customers

Chapter / 05

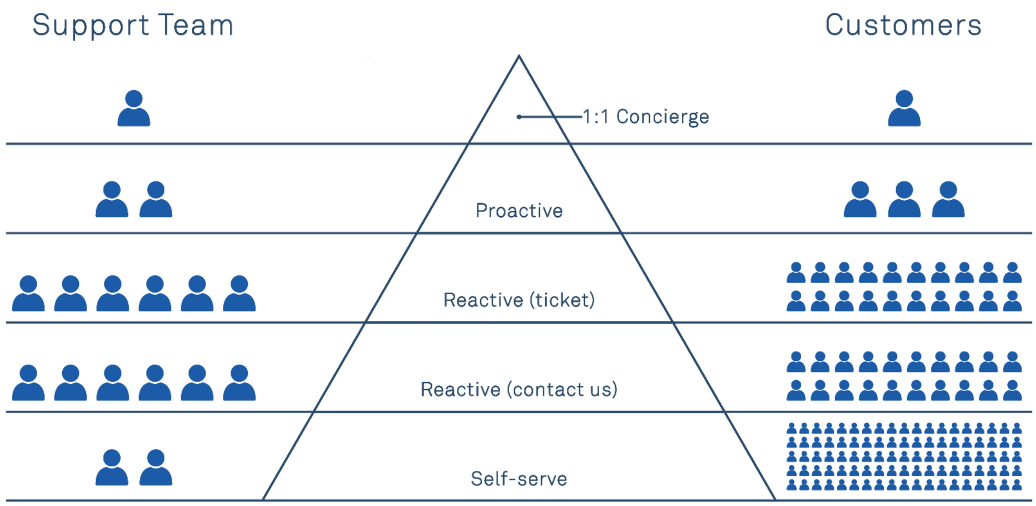


No matter how good your support team is, it's an inescapable fact that some customers simply don't want to talk to them. Perhaps the issue they are trying to resolve feels so simple they just want to be quickly pointed in the right direction. Maybe they've had bad support experiences in the past (with other companies, naturally) and don't fancy hanging around for a response. And then there are some people who just prefer to learn things for themselves rather than have the answer spoon-fed to them.

Whatever the reason there's an expectation on behalf of customers that they should be able to easily find useful and relevant documentation to help themselves. And in our "always on" world, it's increasingly apparent that 24-hour access is becoming table stakes - regardless of the kind of product you are providing.

Your self-service content may take the form of FAQs, best practice guides, very focused how-to's, or setup guides for new users. It might be text only, rely heavily on screenshots, or employ gifs and videos. It can live on a website in a traditional knowledge base or you can insert it into conversations with your customers at the time they are requesting help. The type of self-service you provide isn't just a matter of personal preference. It will have a significant impact on how much time, money and other resources you have to invest in maintaining self-serve resources like documentation, videos and screenshots.

Efficiency of different forms of support



Not only does self-service content allow you to provide great customer service at a large scale, but the cost of providing such service is much lower than every other channel of support.

Don't believe us? Let's take a look at the numbers.

Customer Service Channel	Cost Per Contact
Call center technical support	$12 and higher
Call center support rep	$6 and higher
Web chat or callback	$5 and higher
Email response	$2.50 to $5 and higher
Self-service	$0.10 or less
Source: Forrester Research	

Let's begin by looking at the jobs help content does for you, your company and your customers.

What jobs does help content perform?

1. Enables customers to help themselves

A help center is the ideal medium to help customers with a specific question get a quick answer. So many support queries come from new customers who don't necessarily have a "problem" but are simply confused about a particular issue. Self-service provides new customers with a great tool, and also attracts the natural DIYers.

Need to reset your password but don't know how to do it? Clearly a concise document describing the process, which is surfaced at the correct moment for the customer, is the ideal way to answer this kind of question.

No matter how efficient your support team or how quickly you respond to questions, these kind of low value transactional queries are best handled in a self-service format. The name of the game here is guiding the customer from question to answer in as speedy and efficient a manner as possible. There's usually little value for you or the customer in making the interaction any more complex.

HOW TO MEASURE SUCCESS?

Firstly, are your users finding the documentation when they need it? e.g. did the number of contacts regarding password resets decrease after you created a document about it? If not you might want to consider:

- Macro issues e.g. have you structured your help center in the optimum manner? Are new users easily able to discover the articles they need?
- Document-specific issues e.g. does the heading on the document clearly reflect the content? Does the document contain the keywords that will ensure it is found when users search for it?

The ultimate measure of success here is direct feedback on the document in question. If users are providing positive feedback that their issue has been resolved – whether that's in the form of a thumbs up/smiley face, comments on the article, or conversations initiated from that page – then you are succeeding at this job.

2. Provides an environment for continuous learning

In *Exceptional Service, Exceptional Profit*, Leonardo Inghilleri and Micah Solomon explain why reactive customer service is on its way out – it's an ineffective way to create loyal customers.

"The magic happens when your business anticipates the needs of your customers, learning to recognize and respond to the needs of your customers before they are expressed – sometimes before your customers even realize they have a need." Help centers are the best way of anticipating problems in the first place, giving customers the materials they need to get to grips with your product.

People come to your help site with the intent of having questions answered and learning more about your product. Only answering a single question at a time when they are positively disposed towards learning more is a huge wasted opportunity. The power of customer intent should never be underestimated: remember, Google built a multi-billion dollar advertising business because people searching for products or services are considered to have a far greater intent to purchase them.

HOW TO MEASURE SUCCESS?

You should see an increase in pages per session as you build out the amount of documentation you provide. If all your visitors are coming to your site, reading one article and bouncing again you need to consider how you can improve your current articles and expose visitors to more of your content.

3. Promotes best practice

If you're familiar with Intercom's help content, you'll know we are big believers in providing educational and motivational help content. It's been our experience that instead of pointing customers to a threadbare FAQ or providing them with a series of functional how-tos, creating best practice content leads to loyal and engaged users. We work hard to uncover real-life examples, expert tips and insider secrets so our customers don't have to.

An example of this in practice was creating a guide to writing engaging message content, rather than a series of how-tos around our Engage product. Best practice guides like that give us a chance to educate customers and make them more successful with Intercom.

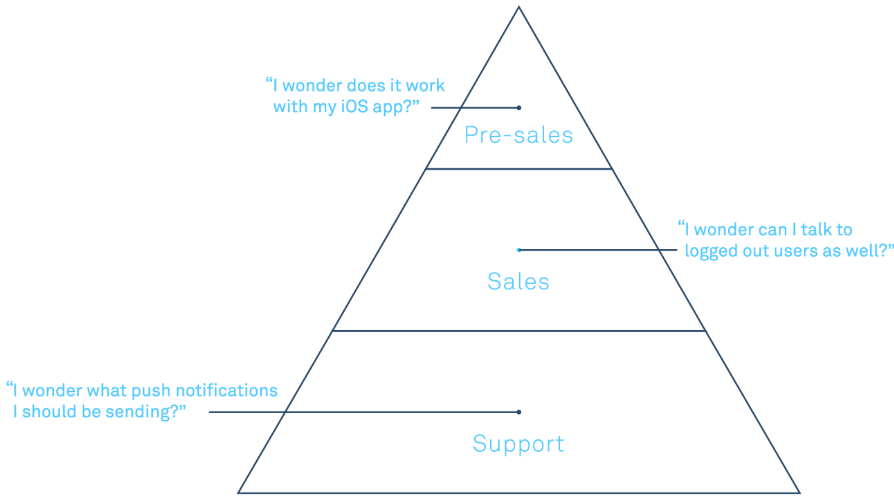
HOW TO MEASURE SUCCESS?

The success of best practice content is hard to measure quantitatively. You should certainly expect to get positive customer feedback - whether that's solicited (e.g. surveys) or unsolicited (e.g. communicated to your frontline support staff). If your best practice guides are performing effectively then you should also see increased product usage e.g. if Intercom produces a guide to sending the right message, to the right user at the right time, then we'd like to see increased use of our Engage product.

4. Sells your product

Most startups assume their help content will only be read by people who already use their product but are having a difficulty with it. But there's another really important constituency self-service content addresses - people who are considering buying it and want to learn more.

Think about the possibilities of your help center
at every step of the customer journey



For example they may have seen it in use on the web but want to know if you have an iOS client. Or maybe they want to know what level of customization you provide. Or which of the existing tools they use you have off-the-shelf integrations with.

This will be especially true if you have a fundamentally different approach to solving a problem compared to the products you compete with. If that's the case, you might want to consider creating documentation that addresses questions people are likely to have when they are assessing your product.

HOW TO MEASURE SUCCESS?

If this strategy is successful your sales and support teams should have to deal with less of these queries. And if you've created sales-specific content you'll want to see salespeople using those articles to help close deals.

5. Raises your visibility

According to Pew Internet Research, 92% of adults use search engines to find information on the web. So where is the first place your customers will turn to for help with your product? The search bar in their web browser. It's essential that you own the problem and that you are answering customer issues. The alternative is well-intentioned but possibly ill-informed users on third party forums, or worse still your competitors. That's why it is so important your help site, at a bare minimum, follows the basics of good SEO. This also helps your content to be discovered in the pre-sales stage as discussed earlier.

There's a whole other book to be written on the topic of SEO but suffice to say you need to get the basics right to ensure your help content is discoverable e.g. use clear headings and titles, include the words that site visitors are searching for but don't engage in shady tactics like keyword stuffing, and remember that ultimately high quality content will win out over time.

HOW TO MEASURE SUCCESS?

Does the organic search traffic to your help site increase after you add new docs or optimize existing ones? That's a pretty simple query to run in Google Analytics.

What type of help content should I create?

There are different types of help articles you can create to address different problems, but broadly speaking there are five kinds of articles you'll need to publish. Because not everyone learns in the same way or at the same pace, a good help site will need to have a mix of article types. And different types of issues also require different kinds of articles to address them.

For example, you could create an FAQ to answer a question customers often ask but which has a relatively straightforward response, while a how-to guide is more suitable to help them navigate a tricky workflow. Help content for new customers needs to be to the point without assuming prior knowledge. But best practice guides can be more in-depth as they are intended for more experienced users trying to get more out of your product.

Getting started guide

Brand new users are unfamiliar with your product and are going to have questions about setting up. Your getting started guide should show your customers a few, quick, valuable ways to get started with your product.

- Focus on the most important aspects of getting started, like installing, configuring, etc. and don't include information that you only need in week 2 or 3 e.g. keyboard shortcuts or account settings.
- Keep instructions short so customers can get set up right away and start getting value out of your product.

- Consider complementing your guide with a short video to walk customers through key features and workflows.
- Are there "quick wins" you can nudge users towards which are likely to make them more engaged with your product?

Best practice articles

Best practice articles should help your customers excel at the job they hired your product for. They will be less product-specific than other types of help content you are going to create. For example, to help our customers succeed with Engage, we created a guide to email marketing best practices. This focused mainly on how to write engaging content regardless of which tool you actually use to send it.

Start by sharing tips on how *you* use your product, but also ask your team, and even your customers, for their best-practice tips.

- Share real-life customer examples (not abstract use cases customers can't relate to).
- Interview your team to discover their expert advice for how they use your product.
- Ask your team product-related and non-product related questions, like "What job does this feature help customers do?" and "If this feature didn't exist how could customers excel at this job?"

Pro tip: Your article title and subheadings should focus on the job your customers want to do, not the feature they want to use. For example, "Tracking your project's progress" is a good title for a best practice guide but “How to use ExampleApp’s tracking feature” isn’t.

How-to guides

If you’ve got a complex product, or an important but tricky workflow, document the steps for getting through it. For example, if installing your product requires technical skills, you should create a step-by-step article to help customers when they hit a problem. Here’s how to identify and address your customers’ pain points:

- Talk to your customer support team (or better still, talk to your customers) to find out where they’re getting stuck and prioritize creating content for those use cases.
- Walk through tricky workflows yourself so you can create content based on your own experiences as a user. It’s easier to have empathy for your customers when you’ve experienced their problem first-hand.
- Ask a teammate to walk through the workflow too – they might spot pain points you’ve missed. This also helps you eliminate any bias in your writing.
- The trickier a workflow is to navigate, the more important it is to use simple, clear language to explain it.

Frequently asked questions

Create FAQs to try and proactively answer common questions customers ask. For example, you could outline a workaround for an issue many customers are experiencing. Or you could explain why you haven’t built a frequently requested feature. Here’s how to create useful FAQs customers are likely to search for:

- Ask your customer support team to tag questions your customers often ask. Then when you write an FAQ about your calendar feature, for example, you can search for your “Calendar FAQ” tag to see where your customers’ frustrations lie.
- If lots of your customers are asking for a specific feature your product doesn’t have, create an FAQ to explain why you haven’t built it. Provide alternative tips customers can try, with your product or complementary tools, to help them get the job done.
- Keep your answers short and concise so customers can quickly get back to what they were doing.
- Assume customers will search for these answers - include keywords they’re likely to search for in your article title and description to help them find what they need.

Troubleshooting articles

When your customers get stuck using your product they’ll likely search your help center for answers. Identify the most common problems people run into and create a troubleshooting article to help them past each problem.

For example, if it’s common for customers to get an error performing a CSV import, you could create an article that tells them the top five things to try. Here are some tips for creating troubleshooting articles:

- Ask your customer support team to identify the top things that go frequently wrong for your customers. This will help you decide which articles to create.
- Get your hands on any internal troubleshooting guides you can adapt for your help center (your customer support team have probably created these already).
- When writing your article start with the most obvious problems (that most people experience), before narrowing it down to more specific issues.
- To help customers identify what the problem is you should pose a series of questions for them to answer, for example, “Are you running a web proxy?”
- Pay attention to the conversations customers start from your troubleshooting articles - then improve your content based on the actual problems customers are encountering.

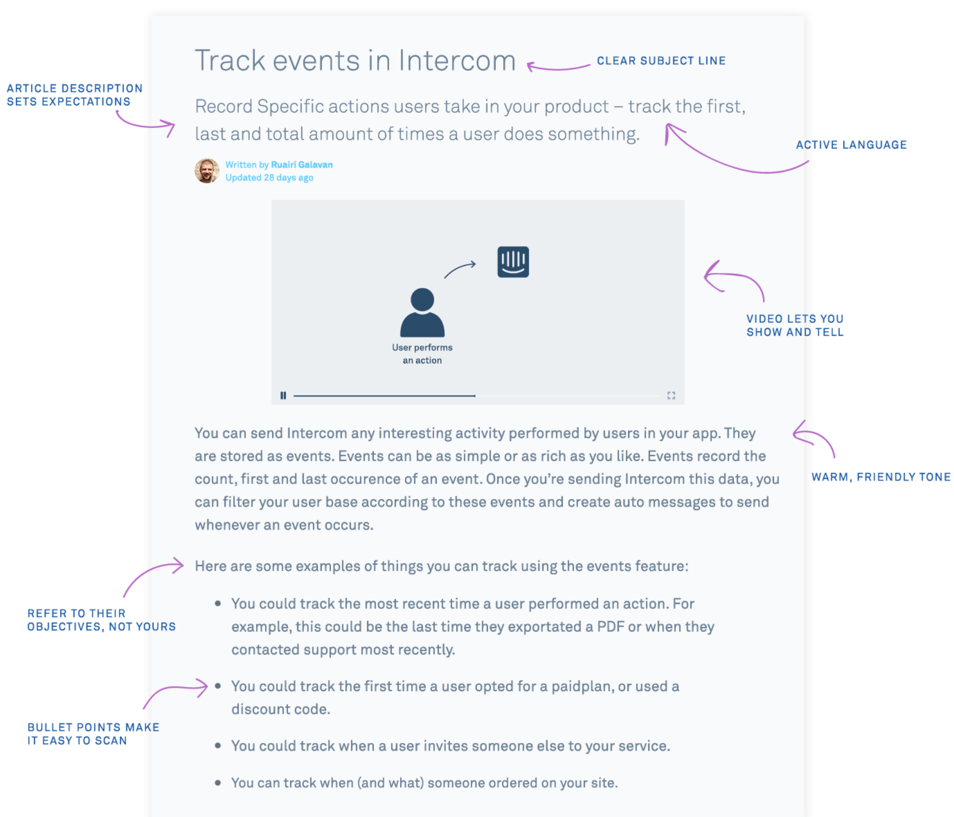
How to write great help articles

We tell ourselves that people don’t RTFM, but trust us – they do. Help articles are usually the first point of contact when someone hits a stumbling block with your product.

So why are they usually so bad? Dry and hard to read, they’re usually organized around the functions of a particular feature, not why someone would want to use it. Often they are designed to deflect customers, at all costs, from contacting support rather than helping them succeed with the product in question.

To create help articles people actually want to read, you need to be accessible, motivational, and mindful of the context your users are likely to be in when looking for help. It's also really important that you consider the reader's likely emotional state when they are visiting your help center. It's quite likely that they might be confused, frustrated or annoyed. Best case they are curious to learn more. All of this influences the tone of your content, which should be straightforward, helpful and friendly. It's probably best to keep humour or the quirky side of your personality for the company Instagram account.

Anatomy of a good help article



1. Define your goal before you start

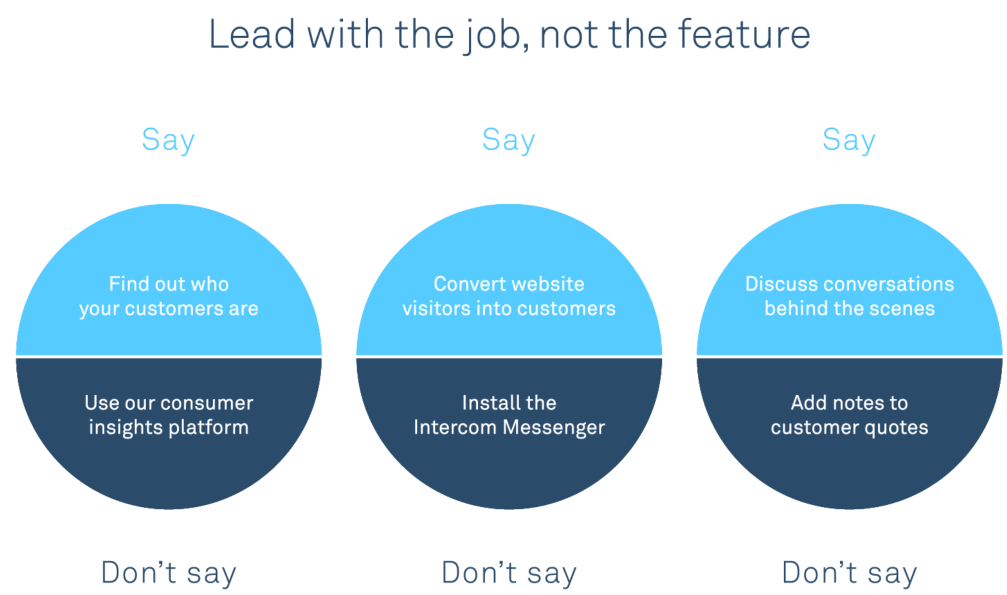
It’s easy to sit down with a blank page and start writing. “Here’s everything someone might need to know using this feature.” Big mistake. Start by asking yourself what someone should be able to do after reading your article, and then reverse engineer your content from there.

No matter how clear your prose, nobody has time to wade through paragraphs and paragraphs of text looking for the nugget they need. So make sure you’re concise and relevant. For example, if you’re writing an article to help your customers get set up, only include tips that provide value right away. Don’t waste precious details before they’re needed.

2. But focus on your customers’ goals too

There’s a reason marketing content is more memorable than most help content – it’s because it has context. Marketing is about the context; help content is about the tool. By adding context to our help articles, we can create content that gives users that ‘Aha!’ moment.

For example, try to focus on the jobs your customers want to do, not the feature they’re trying to use. “Here’s how to work together to reach your project goal on time” is likely to resonate much more than “Here’s how to use ExampleApp’s team feature”.



3. Write using the language of your customers

You'll need to know the words your customers use to ask for help with your product, whether it's on Google or in your help center. So do some keyword research to find the phrases they are using to find your content.

Work those keywords into your:

- **Article title.** Clear, action based titles work best like, "Collaborate together on projects". Short titles will also perform better when it comes to SEO. If you're using Intercom Educate, once you publish your article, the collection it belongs to and the section it's in will also appear in the URL.
- **Article description.** This should set your customer's expectations for what the article contains, and how it can help. Keep it short and to the

point (140 characters max.)

- **Link text.** Writing link text that says “click here” or “find out more” is a wasted opportunity. When linking to related articles, include keywords customers are likely to search for. Here’s a good example: “Check out this guide to collaborating on projects”.

4. Use images and video

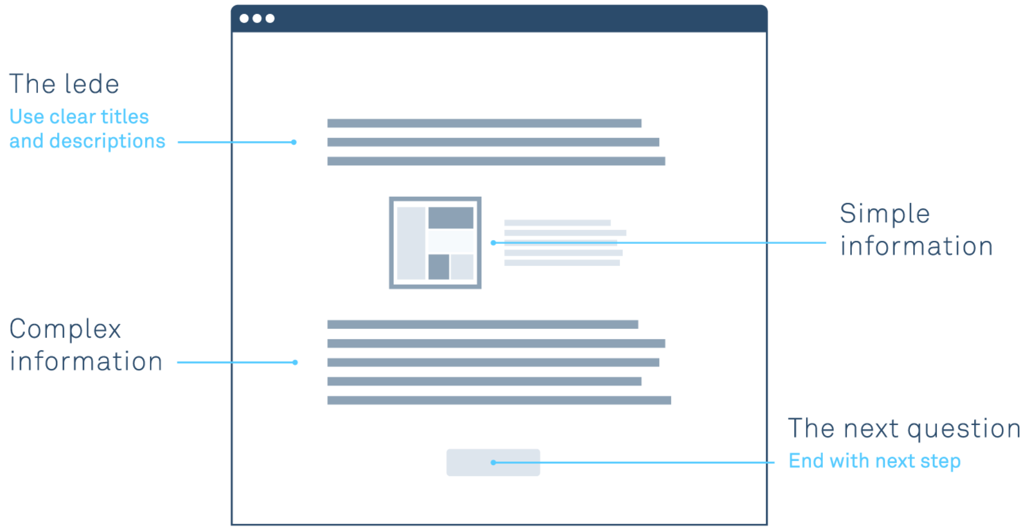
Different people learn in different ways. Simple, bulleted instructions work great for some people, while videos and images will work well for others.

You’re probably familiar with the principle of “Show, don’t tell”. Using images and videos in your documentation lets you simultaneously show *and* tell. They remove the need for enumerating steps or writing overly wordy descriptions, and potentially complex ideas (e.g. how to perform a task that requires multiple steps) can be communicated and comprehended quicker.

5. Add a logical structure

Problems don’t happen in a vacuum, they happen in a workflow of other tasks. The best help articles recognise the workflows of your customers, and have a logical structure that recognises that. Here are a few simple ways you can add structure to your help articles.

How to structure a help article



- **Don't bury the lede.** Within five seconds, a reader should be able to recognise whether the article is useful for them. Don't make them scroll to the bottom only to find out the article was of no use to them in the first place. **Pro tip:** clear titles and descriptions on articles help readers understand what lies ahead.
- **Move from simplicity to complexity.** Users will arrive to your product with different levels of expertise, but it's your job to assume nothing. Before jumping into the nuts and bolts (unless it's a technical article, of course) provide an overview of the feature you're talking about and why the topic you're discussing really matters.
- **Fit it into your product's workflow.** When a customer has a question, you can be almost certain it will be accompanied by a follow up. So try answer it now, so they don't have to go looking elsewhere. That doesn't mean adding every single piece of information about a feature to an article. Use links to other articles customers are sure to want to read about.

6. Make your article easy to scan

Why do so many companies treat potential customers so much better than existing ones? Potential customers get the perfectly formatted landing page, while existing customers get the barely legible block of 10-pt text.

Don't bog readers down with a wall of text – use subheadings, bulleted lists and images to break up text, and keep sentences and paragraphs as short as possible. Remember – white space is your friend.

Maintaining your help content

The days when you could just "publish and forget" help content are long gone. For most software companies, the pace of product releases mean what worked wonders for customers six months ago could be hopelessly out of date today.

More features were added, some features were killed and now new customers arrive and can't make sense of anything. Something that was in "preferences" is now in "settings". New customers don't see this content as being out of date, they see it as confusing as hell.

The only way you'll notice any of these problems is by making your help content something you evaluate regularly. This is hard work, but nobody said helping customers was easy. Follow these steps though, and you'll be heading in the right direction.

1. Put the most work into your most important docs

In an ideal world you'd review and update all your content on at least a monthly basis. But we don't live in an ideal world with unlimited resources. So clearly you have to try and prioritise which content is going to get the most love and figure out which docs only need a periodic review.

The first step is to audit your current documents by plotting them on a graph with pageviews on one axis and value to your business on the other. You should end up with a scatter graph that looks something like this:

If your product is any way complex, the help article on getting set up is going to be one of your most popular (if another article is getting more traffic you probably have more fundamental issues you need to address). And crucially, if users can't get started with your product, your growth is going to stall pretty rapidly so it's of high value to the business.

The highly trafficked articles which are also high value to your business – the ones that occupy the top right on your scatter graph – are also the ones you'll want to consider localizing first as you expand internationally.

Pro tip: Pay close attention to the articles your customers are happiest with. (With Intercom that means the articles they've reacted with a :-) most on). Set a benchmark for success, e.g. 90% positive reactions, and make sure it doesn't fall below that over time.

2. Ruthlessly cull irrelevant documents

Everyone would agree that you should spike older documents that are no longer relevant to your product or business. Maybe the UI has changed significantly. Maybe the feature isn't offered any more. Continuing to provide documentation in those scenarios will just confuse users and end up generating more queries than it solves.

But don't just focus on what's out of date. Cast a critical eye over low-traffic articles too, and review and delete those no longer being visited. Why? Keeping low-performing articles makes it harder for customers to find the content they need and increases your maintenance overhead while delivering a tiny amount of value to the business. The equation is never going to be positive in your favour.

Pro tip: Identify broken or confusing content in Intercom by seeing which customers left a negative reaction on your article and started a conversation from your auto reply. This will help you weed out any broken content or glaring gaps in information right away.

3. Establish a timetable for reviewing content

You should establish a realistic timeframe for comprehensively reviewing all your existing content. Inputs here include the amount of documentation you currently have and how many people you have available to review it.

Here are a few significant events which will warrant a review:

- **When you introduce a new product.** This will obviously impact things like your setup guides and onboarding. Maybe the new product does a job that you previously recommended a workaround for? Are there logical places your current products handover to the new one?
- **When you introduce a major new feature.** Every change you make to your product changes someone's workflow – for better or for worse. Make sure your help articles clearly describe what the feature is and what it does, but also that older articles reflect the new workflow your customers will likely have
- **When you ship a small but valuable feature.** Smaller features might not need you to reinvent the wheel, but they're perfect opportunities to offer more value. For example, a freshly shipped feature could make a useful next step for an older article.
- **When you kill a feature.** Just as your documentation should be constantly evolving so should your product. And if that's happening

you'll need to kill features from time to time. Just make sure your help content is still not referring to them after they've gone.

4. Improve your content based on usage

Seeing what people are actually doing and not doing on your help site provides lots of golden nuggets to help improve your content over time. With a bit of simple analysis you can quickly see which articles to write next, what's popular and most importantly what needs to be fixed. (And if you use Intercom Educate you get all that for free with the Insights feature which also allows you see how your users reacted to each article and the conversations they started from them.)

But regardless of what product you use here's some of the ways you can harness visitor data to fine-tune your self-service content.

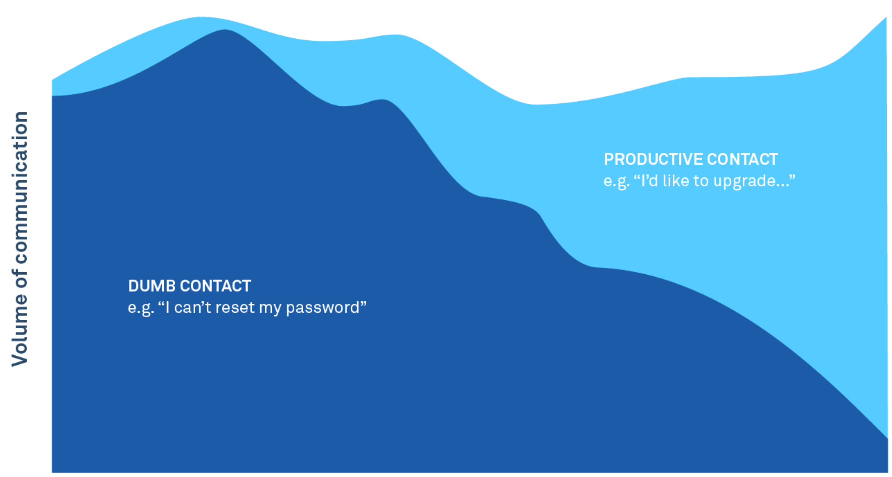
- **Fill in the blanks using search data.** If customers are searching for articles on your help site but don't find any matching content, use the actual keywords customers are searching for in your article title, its description and in the subheadings. These three small tweaks will ensure the next time someone searches for that keyword, it will be one of the first results they'll see. And if lots of your customers are searching for features your product doesn't have, create an FAQ to explain why you haven't built it.
- **Look for recurring trends in feedback.** Getting feedback directly from customers is one of the quickest ways to improve help content. Identify trends in the feedback so you can address issues that confuse customers over time. **Pro tip:** Group the feedback your support team gets with tags, like "Calendar feature questions". When you sit down

to write, you can search for the relevant tag and address the actual problems your customers have.

- **Surface articles that aren't getting viewed.** If an article isn't getting viewed as many times as you hoped, you can create a plan to promote it. Talk to your customer support team and make sure they're sharing relevant content in conversations, and send customers a message directing them to the article, right when they'll need it.

The ultimate goal of self-service

Hopefully you're convinced of the benefits a well written, well maintained help center can have for your business. But remember - letting customers help themselves shouldn't be considered just a cost-cutting exercise.



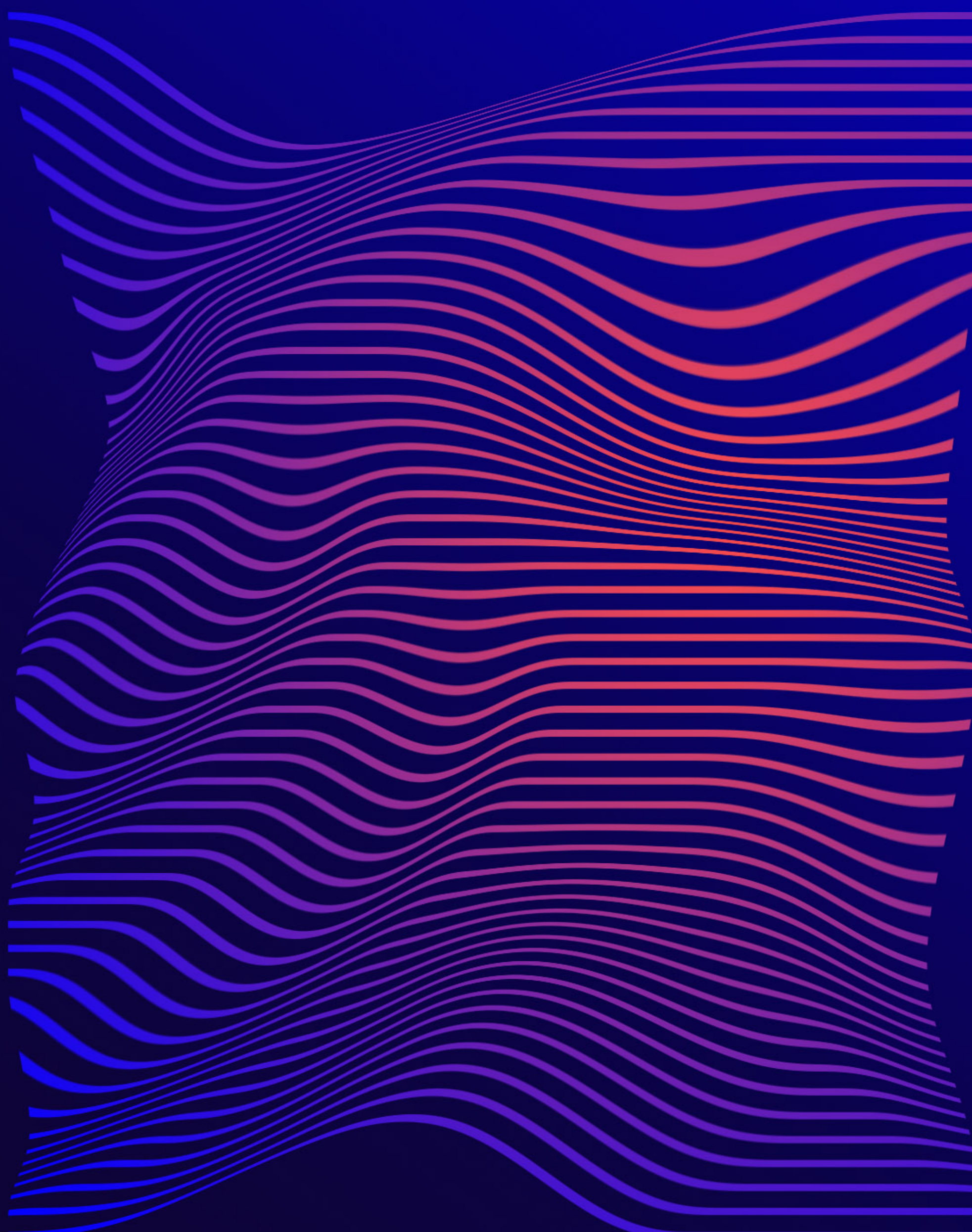
The biggest impact self-service can have on your support operation is enabling your team to focus on the most valuable conversations with customers. We've discussed this already but it doesn't benefit any party if your users are getting in touch with so-called "dumb contacts".

This is not suggesting your loyal users are dumb - but rather the contact is dumb and doesn't benefit either side. Help centers are one of the best ways of filtering out these dumb contacts, and means your support team can focus on the conversations that have the most value to your business.

If you only approach self-service as a means of "ticket-deflection", a low-cost way to reduce your support queues, you'll reduce the overall volume of conversations your support team handles. But you'll never know if you've been missing those deeper interactions which demonstrate your team's value to the rest of the business. More of which next.

Working with the rest of the business

Chapter / 06



It's important your support organization is represented at the top table of your business. It ensures your customers' voices are heard and your leadership team is connected to the customer experience. To make that happen support needs to be considered a peer of the other functions, and this only comes from working closely together every day.

We've already touched on the language you need to employ in order to show the rest of the company the value of support and how it fits into the financial model of your business. But how should you interact with other teams on a daily basis so that support is seen as delivering real value?

We'll cover two examples here:

1. how to create an efficient feedback loop for the teams building your product and
2. how support can help the people who are selling your product.

This is not meant to be a definitive list. Support can also have productive relationships with marketing, growth, finance and any other teams you might have at your company. The examples here should provide plenty of food for thought on how support can start to be seen as a business partner and not just a cost center.

Supporting your product team

It's essential for support to send feedback to the product team. If you outsource support, or even silo your support team internally, you're destined to produce mediocre products. You might pride yourself on being a product-driven company, with an extremely high bar for what you ship. But in practice every product team occasionally ships mistakes. That's going to make your customers unhappy, who in turn make sure your customer support team knows about it. As your company grows and your product team expands, there are more people shipping more things that may be flawed. If there's no feedback loop between your customers and your product team, that's going to result in a lot of upset people.

A product team that is numb to feedback will never learn about their failed launches, their confusing UI, their obscure labels, or their irrational defaults. If they can't be guided by support metrics, alongside product metrics, they'll never hear about what their customers didn't understand.

The damage doesn't just manifest itself in your product, it's in your people too. Designers who fail to learn from their mistakes are destined to repeat them. Engineers who never hear about the impact of their bugs will continue to add more of them. Similarly, an isolated support team evolve into apologists. With no direct connection to the product team they adapt their workflow accordingly. Instead of becoming excellent at communication, customer education, and case analysis, they come up with 20 different ways to say "We know, we're sorry".

Customer support cannot control, or be accountable for, how often customers contact them. They have no leverage there. The majority of contacts for a software product are due to confusion or error. Customers don't understand how to do something, or they can't see why something happened, so they contact support.

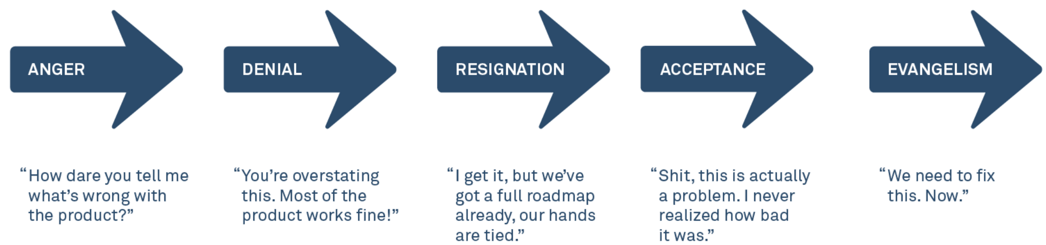
The team is responsible firstly for ensuring that all customers receive a fast, accurate, and friendly answer; and secondly for identifying the product problems that caused those customers to get in touch in the first place and passing them to their rightful owner.

Root cause and owner

CUSTOMER SAYS	ROOT CAUSE	OWNER
"I can't see how to invite my teammates!"	The invite CTA isn't obvious enough	Design
"I can't access my projects!?"	Creating a project with no title crashes product	Engineering
"I was given a discount code but it doesn't work."	Old coupon codes still being handed out	Marketing
"You guys don't explain anything. Where can I find help?"	Customer not seeing docs during sign-up	Design
"I'm still waiting to hear back about this issue!"	Support dropping customer tickets	Support

Every problem has a root cause, every root cause has an owner. A customer couldn't work out how to sign up? Why not? A customer asked how to add his teammates? Why did she need to ask that? These all have owners, people who have the authority and ability to change the product to ensure future customers won't ask this question again.

The five stages of owning your problems



When faced with their problems, product owners often go through five stages of grief. Attempts to dispatch, deny, discredit, or deflect the data will always precede action.

It's a natural reaction. The quickest way to accelerate through it is either for the product owner to sit in on some user testing, or start reading and answering support requests. Amazon solved it by automatically BCCing owners on every single customer issue until they started solving them. As with any internal problem, when all else fails, go for the jugular: the inbox.

Implementing a feedback loop

To start creating this feedback loop you need to record and log all queries from customers with their root cause. Then you can see which features or areas are most commonly generating user questions, and the relevant product managers, designers, and engineers can quickly see the implications of their work. It can also be useful to record the point in the customer lifecycle when the issue occurred e.g. you can see the most common "first day" questions and feed that back into onboarding.

Each month our support team receives tens of thousands of messages from all different types of users - non-customers, new customers, free customers, VIP customers, lapsed customers, big money customers, etc. Although it adds to the support team's workload, every one of those messages is recorded when the customer first gets in touch. As the support team grew it would have been tempting to say this was too much work and trust product managers to occasionally monitor the feedback relevant to their product. But consider the value we would have been throwing away.

Feedback that comes through your support team helps you find the unknown unknowns. It is workflow driven, comes through open conversations, and is limitless in scope. It is about problems you'll never survey about because they're in your blind spot.

The best customer support teams are extremely efficient in everything they do but particularly so with their use of customer feedback. Wasteful teams will just solve the customer's problem, and not consider how that issue can help solve the product problem. In fact many businesses only have strategies in place to solve the customer problem - using tactics like auto responders or automatically suggesting docs or FAQs as possible answers. But if you don't communicate product feedback, it creates trickle down issues: your product gets worse and your customers get unhappier. You're putting band-aids over bullet wounds, and before you know it you have a bad product and a bad product team, when recently you prided yourself on being a product-driven company.

From feedback to action

So you have all these interactions and conversations with customers that contain important nuggets of information. How do you ensure your product teams get this information? How do you categorize and organize, and essentially weaponize, those conversations so the product team can take appropriate action?

Without good data you'll fall victim to the "fre-cently" bias – you assume things you hear frequently or recently have the most impact and so need to be addressed first. But diligently recording and tagging every interaction you have with your customers ensures what you're passing to your product team is helpful, correct, and backed up by a ton of data.

Keep in mind the ultimate goal of your support team. Sure their job is to answer customer queries in as timely, helpful and efficient a manner as possible. But on a strategic level the goal is to **reduce the need for support**. You should be constantly chipping away at demand instead of just trying to cope with whatever your customers throw at you. Your support team is the one best positioned to gather the data and insights needed to reduce that load. If your support team is not empowered to do this, it will forever be fighting against the tide of customer queries that your bad product choices created.

At Intercom, we developed a set of categories that can be applied to all our customer communications, but are still specific and actionable enough for the product team to find useful. This isn't a trivial task – we iterated several times to find the most effective set. A good set of categories is collectively exhaustive, in that every conversation falls into a category, and mutually exclusive, in that there's no ambiguity between them. Your categories will be different to ours but to make this tangible we'll share what is working for us.

FEATURE REQUEST

Customers always want your product to do more than it currently does. Take advantage of the fact that they are pushing the limits. They will definitely tell you what they want to see without you asking. It's important to let them know the product team will see their request, and to try and understand why they're asking for that feature. Maybe there's a good workaround or an alternative way to solve the problem, or there's a compelling use case that the product team hasn't thought about. Each quarter our research team generates a Customer Voice Report, a list of the top 10 feature requests from our customers, which helps inform our product roadmap for the following quarter.

POSITIVE USER FEEDBACK

This should be relatively easy and obvious to spot, but tag it so the relevant team or teams get the feedback. It's simple human nature to want positive feedback on your work. It's also useful to have positive customer quotes that can be used in marketing materials. Just as important, remember to thank the user for sending it in.

OUTAGE/DEGRADATION

If your site or app is down or unavailable you want to know how wide the impact on your customer base has been. Tagging these communications gives our infrastructure team a sense of how performance issues are impacting customers.

CHURN FEEDBACK

When businesses talk about their churn rate they are usually referring to the rate of cancellations they had in a month. But account cancellation is always a trailing indicator – it's the last thing that happens. By tagging any feedback from churning customers you might be able to take action earlier in the cycle. If a customer is leaving because her own business is failing, that's one thing. But if several customers are cancelling because of the recent feature "improvements" you made, then you've got an opportunity to directly address future churn.

BUG REPORT

Even with the best team, bugs happen when you're making software. By their very nature SaaS businesses are frequently shipping code and features. While that doesn't mean you can rely on your users to do QA for you – they'll tire of that quickly – things will slip through the cracks. When your users spot those bugs and report them, take it seriously. Relay as much information as possible to your team in a clear and efficient manner, thank your customer for taking the time to report the issue, and follow up with them when it's fixed.

CONFUSED

You and your team know your product better than anyone, right? But new users coming to it with fresh eyes and zero context are going to have a different perspective on it. That's why customers will often ask what a feature means or how to use it. Those conversations generally begin something like this: "What does X mean?", "I don't understand why…" or "It doesn't make sense to me that…". This kind of feedback often shines a light on the product team's blindspots.

UNAWARE

Your customers will request or ask about features that already exist in your product. These conversations highlight areas where you need to better communicate what your product can do. If you have an excellent events calendar in your product, but customers keep asking how to record events, then you don't have a product problem, you have an awareness problem.

USER ISSUE

The user requires some help or information, e.g. debugging, which is specific to *their* Intercom usage. This kind of contact is often so user-specific it can be hard to reduce the number of them. Often these issues may seem individually tiny, but if collectively this category causes a lot of conversations, then you should consider more defensive design to stop users breaking their configurations.

PRICING

It’s almost impossible to have pricing that is both simple and effective. As a result you’ll always have pricing questions and feedback (we have tags for both), and you want to make sure the relevant team is getting to hear about those conversations.

DEAL BLOCKER

To support our colleagues in sales, we tag conversations where a user or customer tells us the lack of a certain feature is preventing them from buying an Intercom product or upgrading their current plan. But good product companies don’t build custom features just to close a single deal – that leads to feature creep and market irrelevance.

In addition to the main categories above, we also have a few modifier tags we apply to conversations.

ANGRY

If a customer is upset or feels particularly strong about an issue it’s important you share that information when passing on the feedback.

BILLING

Any conversation related to the billing of a current customer, be it a confusion, a bug, or a simple misunderstanding, gets the billing tag in addition to a standard feedback category tag. This just recognizes the importance to your business of this very specific feedback.

The conversations you're having with your customers should tell you what your users actually want (feature requests), what they currently know (unaware), and what they currently think (confused). For customers, perception is reality, so closing that gap is often a huge opportunity to improve your product. That's improvement you'll be blissfully unaware of if you don't talk to your customers.

Supporting your sales team

There is a natural synergy between the sales and support teams, but often they act in silos. Support keeps closed deals closed, helps the sales teams hit their numbers, reduces churn and keeps the company's revenue growth headed in the right direction. The information that support teams capture can also be critical for companies and sales teams that conduct win-loss analyses. Sales and support teams need to recognize this mutual dependency and accordingly build communication channels that enable smooth collaboration.

When it comes to sales and customer acquisition the rules of engagement have changed utterly. Gone are the days when companies bought software without trying it first. Instead, potential customers want to sign up for free trials, explore product documentation and talk to existing customers - preferably without talking to anyone at your company. While the self-service model has revolutionized the software industry, it's not without its challenges. It's dangerous to assume everyone can work everything out for themselves, to assume they want to work it out for themselves, or even that they are able to do it themselves.

The reality is many "self-service" customers will need your help at different stages during their relationship with you. For example, if you are going to convert them from curious users to paying customers, you need to show them the value of free (which we discussed earlier) and make sure they have an amazing experience. It falls on support to ensure those users quickly get back on track if they run into bugs, confusion or technical snags.

The first human point of contact for many of your self-serve customers will be your support team rather than an account executive or sales rep. Use this to your advantage; few of your employees know your product from top to bottom like your support reps. Answering product questions and resolving issues quickly for new users is fundamental to helping them see the value of your product. Fail on this basic level, and those potential new customers will drop off before converting, or if they do sign up will quickly churn.

Great SaaS sales teams are product focused, and their primary job is to build relationships with potential customers. They do this through a well-defined sales process, designed to identify a lead's core challenges, goals and ideal outcomes, and then communicate how your product can help them get there. Sales conversations are supposed to stay high level and "out of the weeds" of overly feature-heavy conversations. This requires a product knowledge focused on things like use-cases and Jobs-to-be-Done, not features and functions. The key to staying high level is being able to draw on assistance from the support team in answering the more detailed technical questions.

Identifying sales opportunities

An easy first step in building a healthy relationship with sales is learning to identify when a customer should be directed their way. That sounds simple but it is rarely as straightforward as a user requesting to "talk to someone in sales". Usually it means picking up clues that a user needs help identifying the right subscription level, feature set, or custom plan to solve their problem. If these are the kind of user requests you are passing along, sales will be able to do what sales does best and close new business.

The tough part for a team accustomed to answering any and all questions that come their way? Often this process means you don't answer *all* of a customer's questions.

Why? Key to effective selling is the "Give/Get" dynamic. The sales process requires an entry point into conversations about a user's needs and challenges. Having the answer a prospect wants means they will be more willing to give information in exchange for that answer. That is the leverage the sales team needs to start conversations. And if you've got a good sales team in place those conversations will ultimately turn into deals being closed.

Here are some of the common indicators of sales opportunities that appear in support channels:

- "What plan do I need?"
- "I'm thinking about upgrading."
- "How much will it cost if...?"
- "Do you offer...?"
- "Why can't I do X?" (if the answer is "because you don't have the right plan", pass it to sales.)
- "What's the difference between Plan X and Plan Y?"
- "Do you guys compete with XYZ company?"

The important thing to note is these are not "blocker questions" that need to be answered so the user can continue trying out your product. But they are ideal openings for sales conversations.

Now that you know what a sales opportunity looks like, how do you pass it to the sales team? It's pretty straightforward, although doing it skillfully will take a little practice and process management. One of the main factors to consider is likely wait time for a response. If the query comes in at a time when your sales team has bandwidth to respond quickly, then simply pass the query straight away. This gives the contact the seamless experience of talking to a single person from the outset.

There are other occasions where it is better to reply first but copy in sales on the response. You want to let the customer know they would benefit from talking to sales while still meeting expectations around response time. If you sense the answer can't wait, provide it while offering the option to talk to sales. Some users will accept gladly, while others will opt to continue with self-service. Either outcome is good for you because the user continues to be able to evaluate your product, and sales is only passed leads who are interested in talking to them. It's fine – productive even – for sales leads to feel a sense of expectation and urgency. What's not fine is if they feel frustrated.

The way you refer to your teammates in these hand-offs says a lot about your internal relationships, so be thoughtful about that. At Intercom we always keep it personal. Just as we strive to always treat our customers as people, we want our customers to know we see our teammates the same way. So, we always refer to our colleagues and our customers by name, not by title, department, role, or otherwise. "Sure thing, let me pass you to Pete. He'll handle your billing query" and NOT "Let me pass your request to our VP of Sales. He is best placed to talk about our plans for larger customers."

Identifying and passing opportunities to sales requires an understanding of which questions are "answer now" and which are conversation openers. Most queries coming into support fall easily into the "answer now" category - bug reports, questions about features, complaints from unhappy users, etc. Answer these right away.

No matter how sophisticated your automated routing, some sales opportunities will end up in the support queue. Your support reps should feel empowered to quickly pass these along to their peers in sales. If the opportunity turns out to be better suited to a self-serve sale, a friendly redirection from someone on your sales team will keep them on track to conversion.

How support enables the sales process

A great support team helps sales do their jobs better. By adding technical expertise, deeper product knowledge, and customer insights, support enables sales to focus on building fruitful relationships that lead to more upsell opportunities and higher retention rates. There are numerous tactics the support team can employ to make your sales team more productive.

When thinking about how your support team can contribute to sales enablement consider tactics that fall into the following three categories.

1. CREATE SALES-SPECIFIC RESOURCES

For instance, the sales team's FAQs relating to a specific feature should be more streamlined than your full product docs and address the questions customers ask in the sales process.

2. DEEPEN PRODUCT KNOWLEDGE

Your support team can train your sales team to conduct more in-depth product demos, answer specific questions faster, and ultimately, help them become domain and use case experts.

3. MAKE SERVICE PERSONAL

Support's participation at crucial moments in the sales process can make all the difference to sealing a deal. Joining calls to help answer specific product questions or technical concerns can speed up the sales process and enables the sales rep to stay high level.

Remember: support is everyone's business

This shouldn't be a one-way street with support figuring out ways to help other functions in the business. It's essential everyone in your business appreciates the importance of support and is exposed to the kind of issues that customers regularly face. At Intercom, once a quarter, everyone puts down their normal work and joins the customer support team to answer questions. Engineers or designers don't try to fix things on Customer Day. They report bugs to the correct team, just like they would if they worked on the support team. This is important for creating mutual respect and empathy between support and non-support teams. Lastly, we ask people to write up their high-level feedback of the day and what they learned.

Conclusion

Startups can’t afford to put customer support on the back burner. If you don’t have the right people and processes in place to scale support, you’ll burn out your team, you’ll enrage your customers, and ultimately, you’ll lose their business.

That’s why it is imperative to proactively define how you are going to support users of your product. You need to decide what level of support you provide, when you provide it, and in what languages.

These decisions establish the parameters of your support organization. Assuming your product team doesn’t drop its quality bar, it becomes relatively straightforward to plan for and scale support experiences that meet or exceed customer expectations.

Providing good customer support costs money – there’s no avoiding that. Don’t think of support as a cost center, but amongst other things, consider it an investment that increases customer retention and reduces churn. And ensure your support team is constantly looking at ways to interact with, and add value to, the rest of your business.

We can’t tell you exactly how you should provide support in your business, but we’ve defined a framework so you can make your own decisions about the specifics. We’d love to hear what works for you. If you haven’t already, check out our blog, Inside Intercom, where we share our latest thinking on customer support and a range of topics relevant to internet businesses, such as product management, design, and marketing. You can also email us at team@intercom.io, or follow us on Twitter or Facebook.

If you liked this book, we'd love for you to share it with your friends and colleagues:

Facebook | Twitter | LinkedIn

Why not download our other books?

Our previous books have been downloaded and read by people at companies like Google, Dropbox, Github, Twitter, Facebook and New Relic.

PRAISE FOR INTERCOM ON ONBOARDING

"Onboarding means ensuring as many users as possible become successful ones. This book outlines time-tested thinking that will help your company do exactly that." – **Samuel Hulick, founder UserOnboard.com.**

PRAISE FOR INTERCOM ON JOBS-TO-BE-DONE

"Jobs-to-be-Done is the light that guides your product through the abyss. This book shows how you can apply it to any organization." – **Bob Moesta, CEO, Rewired Group.**

PRAISE FOR INTERCOM ON PRODUCT MANAGEMENT

"Consistently brilliant writing that should be on the "must read" list for anyone driven to build great products." – **Dharmesh Shah, Founder/CTO of HubSpot.**

PRAISE FOR INTERCOM ON CUSTOMER ENGAGEMENT

“Engaging your customers in a meaningful way is the key to any customer-centric strategy. Intercom is at the forefront of meaningful customer communications, making this book a must-read.” - **Bill Macaitis, former Chief Marketing Officer of Slack.**

How Intercom can help

If you want to provide personal customer support at scale Intercom has two products that can help:

- **Educate** lets customers help themselves with a knowledge base that constantly improves
- With **Resolve** you can support customers via in-app chat, email and social with a live help desk

Used together they offer a complete solution for customer support:

Help center

A beautiful knowledge base for customers to help themselves

Multi-channel

Let customers get help inside your apps, by email, and via social

Team inbox

Respond to customers across all channels from a team inbox

Actionable insights

Measure performance, spot trends, and know how to improve

Try Resolve and Educate free for 14 days