

THE ULTIMATE GUIDE TO ARR

What we've learned building models and standing up ARR reporting at the likes of Intercom, Atlassian, Stripe, and now, Equals (using Equals).

Equals combines the flexibility of a spreadsheet and the power of business intelligence. It's the easiest way to query, analyze, and report on data – wherever it lives.

We also share stories and ideas about building and scaling startups on our blog, <u>Wrap Text</u>.

Written by: Bobby Pinero and Chris Burgner

Edited by: John Collins and Matt Hodges

Design and illustrations: Ben McRedmond and Martin Rariga

© 2024 Equals Inc. ISBN 979-8-9904248-0-7

We'll spare you the badly written, slightly scary-sounding legal warnings. In return, we ask that you give us appropriate credit and a link if you share this book, quote the content, or use the imagery in it.

Got questions? Drop us a note at help@equals.app

This book is dedicated to all the first finance and data hires. And all the friends of finance too. May the ARR be with you.

Bobby and Chris, <u>Equals Experts</u>

Genesis	6
Introduction	8
1 – ARR as Your North Star	10
2 – The Components of ARR	14
3 – Cohorting ARR	25
4 – Defining Your ARR	31
5 – Building ARR	44
6 – Acting on ARR	57
7 – Benchmarking ARR	64
Taking Control	69

Genesis

I first met Eoghan McCabe and Mamoon Hamid back in 2013. I was 25 years young and only three years into my career, but I was raring to help build up an early-stage company. In hindsight, I had no idea what I was getting myself into.

Intercom—the company Eoghan was CEO and co-founder of—needed help. A mere 18 people strong, they'd just raised an exciting Series A from Social Capital, and as a result one of the biggest name investors in the Valley, Mamoon Hamid, had joined their board. As the story goes, Eoghan showed up to that first board meeting with a deck highlighting the product—what'd gone well, what needed to be fixed, and their roadmap. He only covered the product.

Intercom's founders were product visionaries, and the company quickly became one of the shining examples of what it is like to build a product-first company. Arguably, Intercom was one of the first examples of PLG (product-led growth) in SaaS. Yet, something critical was missing from that board meeting. How was the business doing? What was working? What wasn't? Where was the business pointed?

Enter me, the first finance hire. I was brought in to align the company and investors on the story of the business. I had no idea where to start, and few resources existed to point me in the right direction. But I knew the foundational questions we needed to answer were deeply rooted in what was happening with Annual Recurring Revenue, commonly referred to as ARR.

We needed to explain how and why we were growing, uncover opportunities to grow faster, and clearly articulate where we might ultimately end up. It was clear that if we could nail ARR reporting, all the answers would flow from there.

So began an almost 8-year journey marked full of learnings, massive wins, and a lot of mistakes—many of them avoidable. We had transformed Intercom from a <\$1M ARR startup to a \$150M+ ARR juggernaut by the time I left.

Looking back, I wish this book had been sitting on my desk waiting for me on my first day at Intercom. It's co-written by Chris Burgner, who joined me on the Finance team at Intercom and now runs Finance at Equals, where he's also built all of our ARR reporting from scratch. Chris also cleaned up many of the messes I made at Intercom as we scaled. Together, we've written a practical guide with a clear set of steps to enable you to set up industry-best ARR reporting, minus all the mistakes we made (the hard way).

- Bobby

Introduction

Over the last decade, we've seen an explosion in subscription-based businesses—from music to TV and even food delivery. We've seen the rise of self-serve and PLG as a motion for selling software. And we've seen the proliferation of Stripe, ETL, and data warehousing tools to support it all. Yet, after many attempts by startup after startup, ARR reporting is still a f****ing nightmare. Pardon the language, but that's how most people who've dealt with it describe it.

Talk to any startup finance hire, and you'll hear the same story. They spend the first several months on the job cobbling ARR together and then spend years holding it together in a brittle system that needs a never-ending supply of duct tape as the company scales.

Out-of-the-box solutions that sit on top of your billing data don't work either. They fail for a variety of reasons, including:

- 1. They're too rigid to create the reporting your business needs. Weak.
- 2. They require you to adapt your systems to fit their reporting. Lame.
- 3. Changes you make to your business break everything. Frustrating.

It sounds painful because it is painful.

Of course, you can BYO (bring/build your own). We prefer the get-it-all-into-a-data-warehouse-and-transform approach, but it's really hard if you've never done it before. While Finance teams own ARR reporting, many don't know SQL and have no idea how to structure a table, let alone set up a resilient system. It's a recipe for a hot mess.

This book will help you avoid making that mess yourself. It's written for the first finance or data hires at early-stage SaaS companies. The people who know why ARR reporting is so fundamental but need to learn what great looks like. People who want to adopt best practices for setting up systems so reporting on ARR is easy, scalable, and resilient to changes in their business.

It's also valuable reading for anyone inheriting an existing mess of ARR reporting. Godspeed. Finally, for founders and CEOs, this book makes a great gift for your friends in finance. Leave a copy waiting for them on their desk (or email inbox). Heck, you might even want to read it before their first day. \bigcirc

CHAPTER 1

ARR as Your North Star

BY BOBBY PINERO, CEO AND CO-FOUNDER

You don't have to explain to a local restaurant that it will live or die by the number of paying customers it serves. You don't have to explain to an airline that its ability to stay operational wholly depends on its ability to sell airline tickets. You don't have to convince a car dealership that it needs to sell cars.

Yet, for some reason (thanks venture capital!), you have to explain to technology founders and their Finance and Data teams that revenue matters. Believe it or not, some would tell you otherwise.

"This is true in good times, and it's especially true now, during moments of market uncertainty. But no matter how top of mind revenue is for a leadership team, to most people inside a startup, it's not a useful fixation. And as data teams, it's our job to make sure there's something better to pay attention to."

— <u>Startups shouldn't care about revenue</u>, Benn Stancil

So, before we go any further, let's establish that ARR is the most important metric in any subscription business. It's the lifeblood of the company, how investors will ultimately value it, and where the rubber meets the road with your customers.

It's the most important metric because it underpins every crucial question in your business. You know how you should always put your oxygen mask on before helping others when you fly? It's the same thing with understanding the dynamics of your business—understand your ARR first before trying to answer other questions. Because if you don't, your company will die. Metaphorically, of course.

Nailing <u>ARR reporting</u> will help keep your business breathing. It's the foundation for answering nearly every other question in your business. Practically speaking, it's the very first thing you'll want to segment by when asking those questions.

For example, as we started building the world's first "connected spreadsheet", we wanted to know, "What data sources are users connecting Equals to?". The next, most interesting, and most important question we wanted to answer was, "Of the users who connected data source X, what percentage were paying for Equals vs. those that connected data source Y?". And from there, "What plan are they on?", "How much do they pay?" and so on and so forth.

Still not convinced? OK. Let's say you're trying to figure out which lead sources or marketing channels are working. Your first and most important question will be whether those leads convert to paid customers. If so, for how much? And how long do they stick around?

Need one more? Too easy. When determining how to measure <u>onboarding</u> and define user activation, you'll want to ensure that whatever you're looking to drive users to do in your product correlates nicely with them becoming a paid and retained customer.

You get the picture.

Now, here's the thing: Your ability to set up the right metrics—funnel and

marketing channel performance, product engagement and activation, and everything in between—is entirely dependent on having rock-solid <u>ARR reporting</u> in place. Get it in place first so your business doesn't sink.

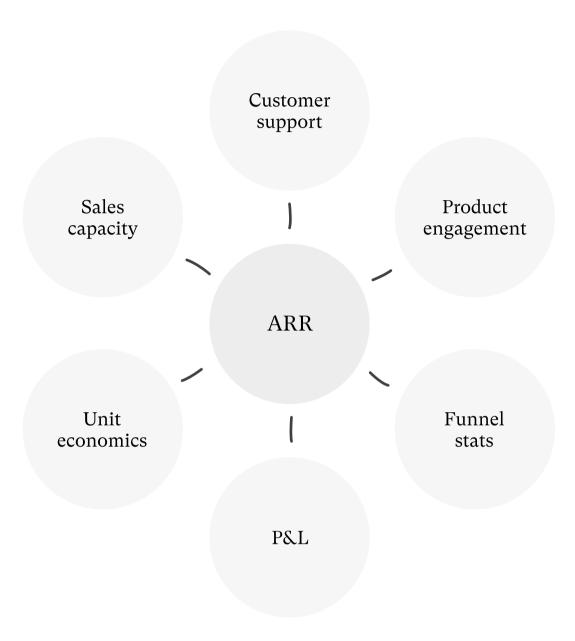


Figure 1.1 - To measure anything, you must be able to measure ARR

With that out of the way, here's what you can expect in the chapters to come. We'll show you:

What good great ARR reporting looks like

Drawing on our experience, we'll explain how to break ARR down into its component parts in order to connect reporting back to the teams that can influence outcomes. We'll also cover the various ways we like to look at cohorts, and lastly, how we think about forecasting ARR.

How to work through the most challenging set-ups

From wrangling your data to applying business logic and setting up a source of truth, we'll give you practical guidance to consider. Every business is different, so we don't have hard and fast rules. No single book can tell you exactly what to do but we'll give you a framework to apply. If you need more specific help or want us to do it for you, we'll tell you how to do that too.

How to operationalize and make the most of ARR reporting

Once you've established what great ARR reporting looks like and how to set up the underlying infrastructure, you'll be ready to share your company's view on ARR with your broader team. We'll share some of our best practices to ensure that it's a success.

What benchmarks for (healthy) ARR growth look like

Overall, we tend to be skeptical about benchmarks. They're dangerous when applied without context and very specific filters. That said, we provide some high-level benchmarks surrounding ARR and connected metrics that have proven helpful for us.

And, one more thing...

We share a curated collection of our favorite and most valued resources on the topic of ARR.

Sound good? Then let's jump in!

CHAPTER 2

The Components of ARR

BY CHRIS BURGNER, HEAD OF FINANCE AND ANALYTICS

Hi, Chris here. You know, the guy who cleaned up all of Bobby's messes when we worked together at Intercom? That one. Let me start with a story he told me when I was cutting my ARR teeth. For that, I'll quickly adopt Bobby's voice.

It was my first week at Intercom, and a report from one of our investors at Social Capital landed in my inbox. It was the first time I'd seen an ARR Build (see Figure 2.1).

Ever been on a blind date? I have. It was like that—very daunting.

The ARR Build didn't make much sense to me

. I didn't have

a clue what it meant to record "Gross New ARR" or how to think about "Expansion", "Contraction", or "Churn." And what on earth was "NRR"? So. Many. Acronyms.

But what was scary at first quickly became clear as I asked questions and listened—the recipe for any successful date.

I quickly learned why breaking ARR into its component parts was critical to understanding the story of the business and how to influence it. Each component represented a distinct and different area of the business we could impact. With this fundamental and deep understanding of Intercom's ARR, we oriented the company. Specific teams were assigned to own different components and their associated targets. This shift in focus played a huge role in driving the hyper-growth in the years that followed.

This "first date" with ARR marked the beginning of my journey to unravel what the hell it is, how it comes to be, and how to measure and forecast it properly.

Intercom P&L DRAFT				
7/31/2013				
	4/30/13	5/31/13	6/30/13	7/31/13
MRR				
New MRR				
Churned MRR				
Expansion MRR				
Net New MRR		100.000	10.000	8.00
Starting MRR			1000	
Ending MRR		1000	1071,000	1900,000
% growth MoM				
Implied discounts given				
Gross MRR				

Figure 2.1 - To understand ARR it should be broken into its component parts

Isn't Bobby a great storyteller?

Alright, let's get right into it. Starting from the ground up.

Total ARR

Reporting on ARR is like building a house: You need to start with a solid foundation. That foundation is Total ARR, which is a great place to start because it's easy to measure and calculate. Put simply, it's the total Annual Recurring Revenue from all customers over time.

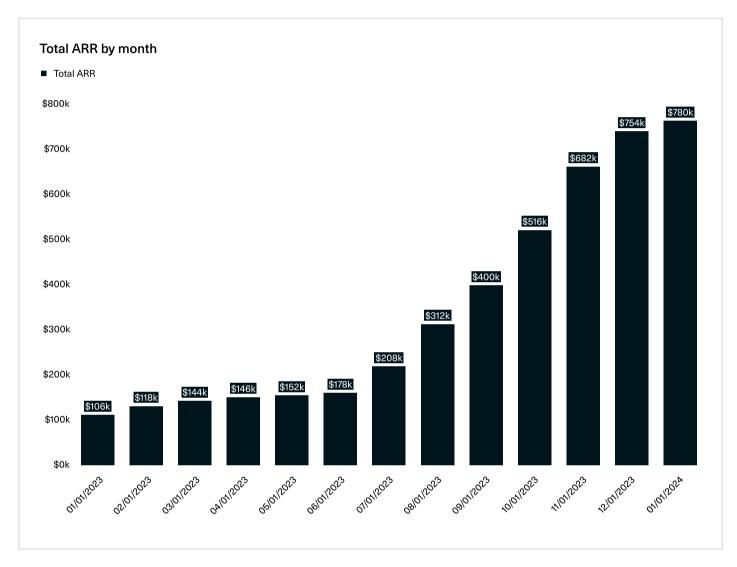


Figure 2.2 - Total ARR is the sum of all customer ARR over time

Net New ARR

Next, you want to calculate net changes in ARR from one period to the next, which we call Net New ARR. You'll want to set goals against Net New ARR and examine benchmarks (which we'll discuss later) to understand how well your business is growing.

CALCULATING NET NEW ARR

From here, we need to break down Net New ARR into parts based on the type of ARR, which will expose added and lost revenue in a given period.

This helps paint a clear picture of how customers' actions impact the business. (See Figure 2.4.)

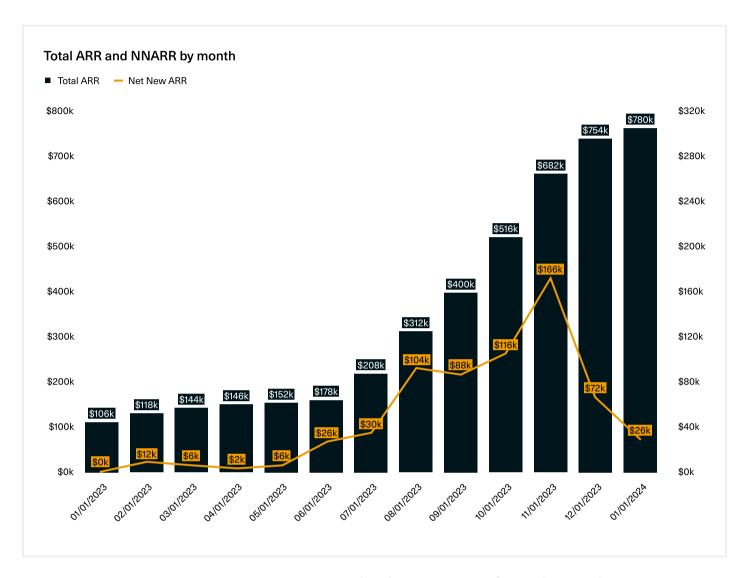


Figure 2.3 - Net New ARR is the change in ARR for each period

The formula for Net New ARR looks like this:

Net New ARR = Gross New ARR + Expansion + Contraction + Churn + Restart

Next we'll cover each component in more detail, including why they are important to the health of your business, how you can influence them, and who should take ownership of that.

Gross New ARR

Gross New ARR comes from new customers who start a paid subscription to your product or service for the first time.

CATEGORY	ARR COMPONENT	EXAMPLE CUSTOMER ACTIONS
Added Revenue	Gross New	Started a new subscription
	Expansion	Upgraded to a higher plan Increased seat count Bought an additional product or feature
	Restart	Restarted previously cancelled subscription
Lost Revenue	Contraction	Downgraded to a cheaper plan Reduced seat count Removed add-ons
	Churn	Cancelled subscription Downgraded to free plan

Figure 2.4 - Net New ARR is made up of different categories of added and lost revenue

WHY IT MATTERS

Gross New is crucial in the ARR Build for two reasons:

- 1. When a customer's subscription starts determines what cohort they are part of, which impacts any cohort analyses.
- 2. Gross New makes up the vast majority of Net New ARR for early-stage companies. As the business scales, the balance shifts, and a higher concentration of Net New ARR comes from Expansion ARR from existing customers. This has significant implications for your ability to forecast ARR in the early days—a pain point even the most successful tech companies have had to contend with.

In the early days at Snowflake, Brad Floering, the company's Head of FP&A, realized the accuracy of his topline forecast largely depended on his team's ability to predict the distribution of New vs. Existing customers and the expected usage behavior for those newly acquired customers.

Because this segment of the business was so important in getting the overall forecast right, the company eventually had to create an entire forecast model just for new customers, which they referred to as the <u>Cold Start model</u>.

HOW TO OPTIMIZE IT

Operationalizing this means your sales, marketing, product, and growth teams must be aligned and in lockstep with your go-to-market strategy.

You can help them in this journey by providing clear and concise reporting on the upstream metrics that feed into Gross New ARR, such as <u>top-of-funnel reporting</u> and segmenting performance by firmographic attributes (e.g., geography, company size, buyer persona).

LEVERS TO PULL

- Increase lead volume
- · Improve funnel conversion
- Increase the starting price point

Expansion & Contraction ARR

Everything outside Gross New ARR is influenced by existing customers, starting with Expansion and Contraction.

These components are really two sides of the same coin: one tells you if an existing customer increased their spend (expansion) and the other if they decreased (contraction). The main thing to keep in mind when looking at Expansion and Contraction, and other components that we'll cover next (e.g., Churn), is that in order for a customer's action to fall into the Expansion/ Contraction bucket, they must hold an active subscription vs. cancelling their subscription and no longer be considered a customer for the purposes of ARR reporting. (See Figure 2.5.)

Why it matters

Understanding fluctuations in these components tells you whether customers tend to increase or decrease spend over time, in addition to the consistency in the direction of that spend.

Consider it this way: acquiring new customers is time-consuming and expensive. It often requires a large budget for demand generation, a dedicated new business Sales team, and months of engagement to get a cold lead to buy your product.

Compare that to turning your early adopters into evangelists who promote your product for you. "Word of Mouth" is a beautiful thing. It costs you (almost) nothing in terms of dollars invested, which is why almost every breakthrough startup shows signs of organic growth (with minimal spend on Sales and Marketing), and investors are so keen to track expansion dynamics.

HOW TO OPTIMIZE IT

First impressions are critical to a customer's likelihood of expanding or contracting in the future. Be diligent in tracking <u>engagement and activation metrics</u> as new customers are onboarded. These will be the best predictors of whether a customer will expand or contract in the future.

Work with Sales and Customer Success to roll out customer health dashboards and actively monitor response and resolution times with support. Building the right habits early on will pay dividends down the line.

EVENT	EXPANSION	CONTRACTION	CHURN
Added seats to paid subscription	✓		
Upgraded to a more expensive plan	✓		
Bought an additional product	✓		
Removed seats from paid subscription		✓	
Downgraded to a cheaper paid plan		✓	
Downgraded to a free plan			✓
Cancelled subscription			✓

Figure 2.5 - A change to an existing subscription falls into one of Expansion, Contraction, or Churn

LEVERS TO PULL

Expansion:

- · Pricing that better captures the value customers derive
- · Launch new products and offerings
- · Upsell and cross-sell campaigns

Contraction:

- Improve onboarding and activation of feature sets most commonly dropped
- · Don't oversell in the initial sale

Churn ARR

Churn is usually defined as the point in time when a customer cancels their last remaining subscription. Similar to Gross New ARR, the exact definition depends on where you draw the line on what constitutes a "customer" for the purpose of reporting ARR. But it always involves a cancellation at some level.

WHY IT MATTERS

We've discussed how your business's growth will eventually transition from being driven by customer acquisition (Gross New) early on to being heavier on existing business (Expansion) as the company scales.

Keeping customers active is the biggest lever to speeding up this snowball effect. Companies with a "leaky bucket", meaning a high percentage of customers quickly churn, must work twice as hard to acquire new customers to return to growth. Doing so is resource-intensive (and expensive), which is why investors and operators alike are so focused on Churn performance.

HOW TO OPTIMIZE IT

Like Expansion and Contraction, tracking engagement metrics is critical to understanding the health of your customers and their likelihood of keeping an active subscription.

Investing in teams focused on Customer Support and Success goes a long way to addressing pain points that may lead to Churn down the line. Understanding where in the customer lifecycle Churn takes place can help you infer where help is needed.

Seeing a lot of customers become inactive within the first few months? This might be an issue with onboarding. Alternatively, if you're seeing customers leave once they hit a certain spend threshold or level of usage, that may be a sign their needs are outgrowing your product and investment in additional, usually more sophisticated, features is needed.

Don't wait for things to reach a point of no return before engaging with customers, and make sure you have the right amount of coverage on your teams to serve their needs and be responsive. Investments in Churn-save will likely return many multiples in the future.

LEVERS TO PULL

· Shore up key product gaps/issues

- · Improve onboarding and activation
- Modify pricing and packaging

Restart (AKA Resurrection) ARR

Restarts happen when a customer who had previously churned—cancelled their subscription or downgraded to a free plan—starts a new paid subscription.

Most businesses limit the Restart window to one year. This means that if a customer churns and returns after more than one year and starts a new paid subscription, they will contribute to Gross New ARR. Landing on the right definition for your business depends on your go-to-market motion.

In enterprise companies, where customers are on annual contracts, there's less of a need for a defined window, as the customer proactively communicates Churns in the form of an opt-out. However, self-serve or PLG businesses tend to see customers come and go more frequently and may want to consider adding a 3-6 month limit to the Restart window.

Why it matters

Restart is less important as a performance indicator and more about hygiene. Differentiating between win-back customers and those joining for the very first time helps eliminate false signals around Gross New and leads to better top-of-funnel reporting.

Having said that, understanding why customers come back can provide valuable testimonials for marketing and insights for product to prevent future Churns. It's also a great indicator of the progress you've made as a team, in which you've potentially shored up shortcomings that customers previously felt.

HOW TO OPTIMIZE IT

Because a Restart event mainly depends on the customer taking action, this component of ARR receives less proactive attention.

The obvious tactic is to target win-back campaigns at churned customers. But over the long run, what will ultimately drive restarts is your ability to fix key product issues. Resolve the reasons that customers churn, and you'll see some customers restart.

ARR COMPONENT	DEFINITION	LEVERS
Gross New	The total value of subscriptions from new customers.	Increase lead volume Improve funnel conversion Increase the starting price point
Expansion	The total value of additional spend from existing subscriptions, e.g. add more seats, upgrade to a more expensive plan, buy add-ons.	Pricing that better captures the value customers derive Launch new products and offerings Upsell and cross-sell campaigns
Contraction	The total value of reduced spend from existing subscriptions, e.g. remove seats, downgrade to cheaper paid plan.	Improve onboarding and activation of feature sets most commonly dropped Don't oversell in the initial sale
Churn	The total value of canceled paid subscriptions.	Shore up key product gaps/issues Improve onboarding and activation Modify pricing and packaging
Restart	The total value of new subscriptions from previously churned customers within a given time window, e.g. one year.	Closing known product gaps Persistence

Figure 2.6 - The different components of ARR and how to move them

Summary

Hopefully, you now have a deeper understanding of ARR and each of its components. There was a lot to digest, so we've created a cheat sheet for you (see Figure 2.6 above).

CHAPTER 3

Cohorting ARR

BY CHRIS BURGNER, HEAD OF FINANCE AND ANALYTICS

The components of ARR tell you the direction of the trend at any given time: is it increasing, decreasing, or staying the same?

Cohort analyses help you understand how customers perform as they age and how that performance compares across different customer vintages i.e., cohorts.

OK, but what does that mean, and why is it important?

Cohort analyses are the most straightforward way to see if your business is acquiring sticky customers and whether or not those customers are increasing their spend over time. Demonstrating an ability to do both is the secret to unlocking sustained, durable growth.

This kind of enduring growth has been described in many ways. One of our favorites comes from Jeff Bezos's annual shareholder letter in 2000.

"We're a company that wants to be weighed, and over time, we will be—over the long term, all companies are. In the meantime, we have our heads down working to build a heavier and heavier company."

– Jeff Bezos in his Annual Shareholder Letter (2000)

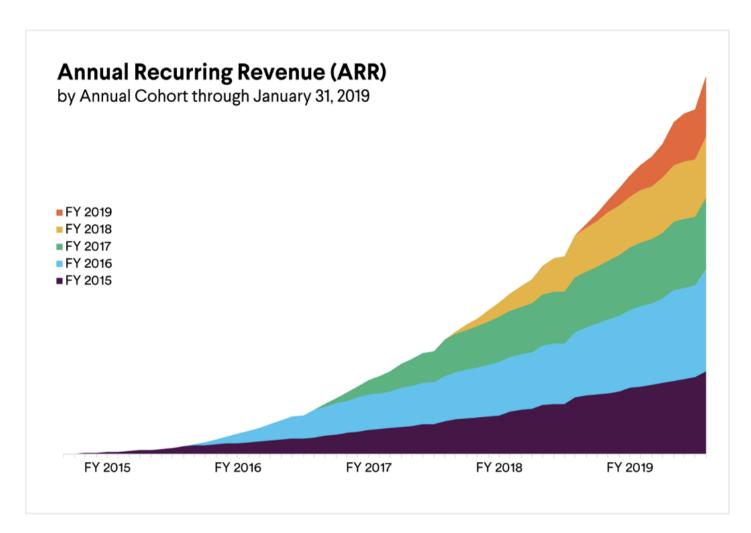


Figure 3.1 - Slack's ARR shows the power of cohorts in accelerating a business's revenue growth

Net Revenue Retention

Net Revenue Retention (NRR) shows how the relative total spend of a customer or set of customers has changed over time.

NRR analyses can be run on a cohorted or uncohorted basis, each requiring slightly different methodologies.

COHORTED

This "left-justified" view cohorts a set of customers based on their subscription start date (y-axis) and compares their total spend across

time (x-axis) to their initial spend (Gross New ARR). The cohorts and aging are typically grouped by months, but as you have more data, it's common to also group on a quarterly basis.

	Total S	pend by	/ Month	>											Cohorte	ed NRR b	y Month	>									
Cohort Month	0	1	2	3	4	5	6	7	8	9	10	11	12	Cohort Month	0	1	2	3	4	5	6	7	8	9	10	11	12
2022-02	\$100	\$102	\$104	\$106	\$108	\$110	\$113	\$115	\$117	\$120	\$122	\$124	\$127	2022-02	100%	102%	104%	106%	108%	110%	113%	115%	117%	120%	122%	124%	127%
2022-03	\$103	\$105	\$106	\$108	\$109	\$111	\$113	\$114	\$116	\$118	\$120	\$121		2022-03	100%	101%	103%	105%	106%	108%	109%	111%	113%	114%	116%	118%	
2022-04	\$107	\$108	\$110	\$111	\$113	\$114	\$115	\$117	\$118	\$120	\$121			2022-04	100%	101%	103%	104%	105%	107%	108%	109%	111%	112%	113%		
2022-05	\$102	\$103	\$105	\$107	\$108	\$110	\$111	\$113	\$115	\$116				2022-05	100%	101%	103%	104%	106%	108%	109%	111%	112%	114%			
2022-06	\$104	\$105	\$106	\$107	\$108	\$109	\$110	\$112	\$113					2022-06	100%	101%	102%	103%	104%	105%	106%	107%	108%				
2022-07	\$102	\$103	\$104	\$105	\$106	\$108	\$109	\$110						2022-07	100%	101%	102%	103%	104%	105%	107%	108%					
2022-08	\$101	\$102	\$103	\$104	\$106	\$107	\$108							2022-08	100%	101%	102%	103%	105%	106%	107%						
2022-09	\$100	\$102	\$103	\$105	\$106	\$108								2022-09	100%	102%	103%	105%	106%	108%							
2022-10	\$111	\$112	\$113	\$115	\$116									2022-10	100%	101%	102%	103%	104%								
2022-11	\$113	\$114	\$116	\$117										2022-11	100%	101%	102%	104%									
2022-12	\$104	\$105	\$106											2022-12	100%	101%	102%										
2023-01	\$105	\$107												2023-01	100%	102%											
2023-02	\$107													2023-02	100%												

Figure 3.2 - An example of cohorted Annual Revenue Revenue (ARR) and Net Revenue Retention (NRR)

Here's how to read the NRR of a given cohort compared to the starting period period (e.g., month one):

<100%	100%	>100%
Relative spend decreased	Relative spend remained flat	Relative spend increased

WHAT TO LOOK OUT FOR

There are two things to be mindful of when looking at Cohorted NRR.

First, how retention compares across customer vintages at fixed age intervals. Said another way, are newer cohorts retaining worse, the same, or better than older cohorts at month 3/6/9? This tells you if the stickiness of your customer base is changing, which could come down to the quality of your product changing and/or the quality of a given cohort of customers.

Second, the slope of retention curves over time should be compared to see if the pace of retention changes with aging cohorts. This will help us understand point-in-time differences and whether lines across vintages converge or diverge over time.

A common case to consider is if you changed pricing or introduced annual contracts. Maybe your initial ACV is higher in more recent cohorts, which results in less relative expansion in the first ~3 months, but the lines converge by months 9-12 as the drop off from Month-to-Month (M2M) customers weighs down the legacy cohorts.

Your business is constantly changing as an early-stage startup. You can think of Cohorts as your map to understand how experiments impact results.

UNCOHORTED

This is the more commonly discussed of the two calculations. It compares the total spend of a fixed set of customers over a fixed window of time, typically 12 months, to show how much ARR is retained. It's important to note that the same set of customers are evaluated, meaning spending related to any Gross New customers acquired in the same fixed window (e.g., 12 months) is not included in the calculation.

The example below highlights the relevant cohorts used to calculate NRR in April 2023:

- 1. Gross New cohorts from May 2022 to April 2023 (red) are ignored from the calculation
- 2. Instead, we're taking the \$381 of "Total Spend (Current)" as of April 2023
- 3. Then, dividing by the \$316 of "Total Spend (OG)" of those same cohorts 12 months earlier
- 4. Yielding an NRR of 121%

Following this methodology ensures the same customers are compared between the April 2023 and April 2022 periods. (See Figure 3.3.)

Almost every public SaaS company reports on NRR in public filings, but be mindful when comparing against peers—we cover why later in Benchmarking ARR.

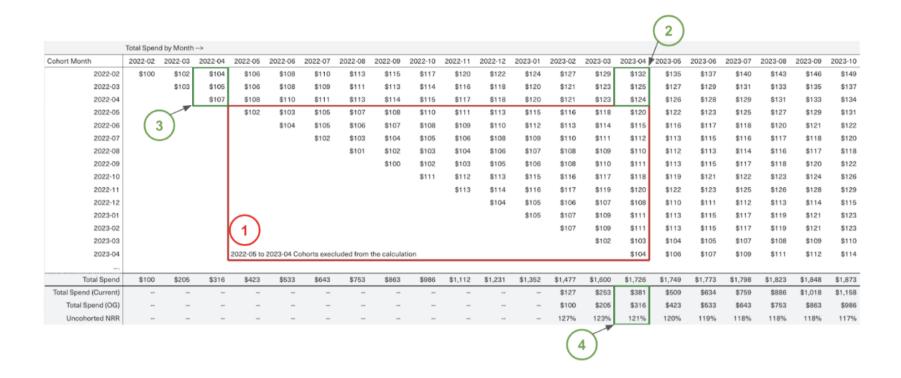


Figure 3.3 - An example of the right-justified unchorted NRR calculation methodology

Gross Revenue Retention

Gross Revenue Retention (GRR) is similar to NRR but only considers Churn and Contraction. Removing the additive revenue from Expansion and Restart gives a clearer picture of how much spend erosion takes place over time.

COHORTED

A "left-justified" view benchmarks customers based on their age, typically in months, and compares their spend at any given time to their initial Gross New ARR.

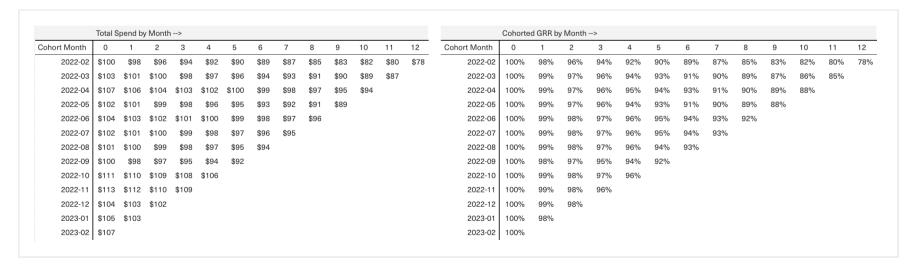


Figure 3.4 - An example of left-justified Total Spend and GRR by month

UNCOHORTED

The "right-justified" view compares the spend of the entire customer base over a defined window—typically 12 months—to understand how much ARR a business should expect to lose from its existing customers.

The example below highlights the relevant cohorts used to calculate NRR in June 2023:

- 1. Gross New cohorts from July 2022 to June 2023 (red) are ignored from the calculation
- 2. Instead, we're taking the \$420 of "Total Spend (Current)", ignoring any uplift from Expansion and Restart as of June 2023
- 3. Then, dividing by the \$499 of "Total Spend (OG)" of those same cohorts 12 months earlier
- 4. Yielding GRR of 84%

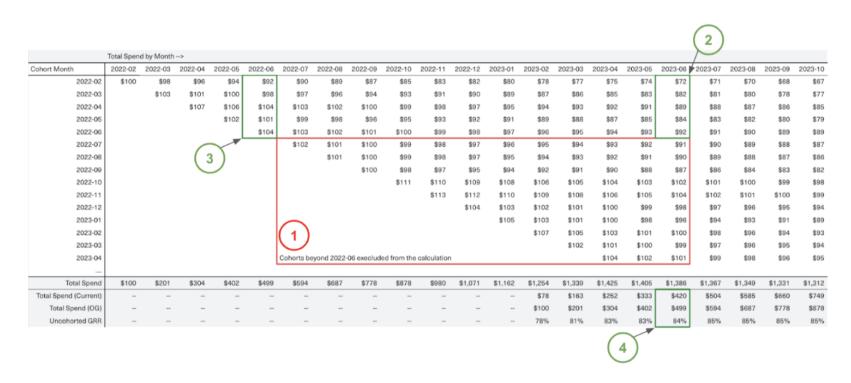


Figure 3.5 - An example of right-justified Total Spend by month

Following this methodology ensures the same customers are compared between the June 2023 and June 2022 periods. (See Figure 3.5.)

CHAPTER 4

Defining Your ARR

BY CHRIS BURGNER, HEAD OF FINANCE AND ANALYTICS

In most SaaS companies, when you reach a certain scale, Finance has its own forecast, and Sales (or Sales Ops) tend to have their own. Your goal should be to align these two groups as closely as possible. Sales and Finance should walk into every forecast review with a tightly aligned view of the state of the business. It took us many iterations (and horrific meetings) to get there at Intercom.

There were many times over the first few months of that relationship when we'd walk into a Go-to-Market forecast review, and Sales would be calling a number millions off from what we were calling in Finance. We looked like buffoons.

This would happen for a variety of reasons. Sometimes, it was because the contract signed date in Salesforce differed from the invoice date. Other times, it would be because a customer would

expand, and Sales would count that, but they'd miss the subsequent contraction event. Another time, a customer might sign a contract but not pay after three months. Finance would count that as a churn, but Sales wouldn't.

We spent months and months reconciling these differences, mainly after the fact, after the painful meeting, and after the embarrassing show of ineptitude in front of the rest of the team.

This chapter is intended to help you not look like the buffoons we did.

As painful as it is to relive, Bobby's story clarifies that having a single source of truth for ARR is critical to getting alignment across your business. How can you expect to take action and move the company forward if you can't agree on where you are?

We'll start by breaking down all the business considerations you need to work through to develop your own source of truth for ARR reporting. While it may seem overwhelming at first, planning this out ahead of time will save countless hours and many arguments debates down the line.

In short, we'll cover the following topics and include recommendations and examples for each:

- Defining a customer
- · Backing out discounts
- · Reporting on service vs. product-based charges
- · Treatment of overages (spend in excess of the contract amount)
- · Building off Subscriptions or Invoices

Defining a Customer

Deciding where to draw the line on what constitutes a "customer" is one of the most critical decisions in reporting on ARR, as it has implications

for cohorting and the component parts of ARR. It's also relevant to how other teams, like Sales and Support, will organize their teams.

CUSTOMER TO SUBSCRIPTION

What to consider: Can a customer have only one active subscription at any given time (1:1) or multiple concurrent subscriptions (1:many)?

What we recommend: Allowing multiple subscriptions to be mapped to a single customer is fine, especially if the buying consideration differs across product lines. Just ensure it's clear in your logic that:

- 1. Gross New ARR is only counted on the first subscription start date
- 2. Churn is logged when the last active subscription is canceled

PARENT-CHILD ACCOUNTS

What to consider: How do you handle cases where you work with multiple subsidiaries that roll up to a single entity (e.g., WhatsApp and Instagram to Meta)?

What we recommend: Any entity with purchasing authority (i.e., can sign a contract) should be considered a customer. We like this definition because it fits well with how you would organize your Sales or Support team, as each entity will likely have its own instance of your product.

You should consider including a "Global Parent" field in your CRM to link sibling accounts to the same parent.

Counting customers

Depending on the buying process, there are several events to choose from as the point at which to start counting a customer for reporting purposes. Remember, this is especially important because it determines which cohort a customer belongs to and impacts operational elements like Sales compensation.

Here are a few things to keep in mind:

TRIAL PERIODS

What to consider: If you offer a trial, do you wait until the trial expires or start counting at signup?

What we recommend: To reduce noise in your reporting and align to the moment where a customer has skin in the game, we suggest recording the revenue once the trial expires and the paid portion of the subscription starts.

PAYMENT STATUS

What to consider: Is getting to trial end and subscription start date enough, or should you wait until an invoice has been paid?

What we recommend: The answer depends on your go-to-market motion.

For companies with a PLG or Self-serve motion, it's better not to count a customer as Gross New until their first invoice is paid. This will mitigate inflating Gross New and reduce immediate Churn from customers where money was never exchanged. This is less relevant for companies with a Sales-led motion, as customers have clearly demonstrated intent to pay when a contract is signed.

CONTRACT START VS. BOOKING DATE

What to consider: If customers must sign a contract before they can use your product, should you start counting them as customers on the signature or contract start date? What if the two dates fall into different reporting periods? It's common for many deals to close on the last day of the quarter, but the contractual relationship begins in the following month.

What we recommend: You'll hear many opinions on this topic, but we advise aligning the ARR recognition to the subscription start date. Doing so will inevitably create a dissonance between your Sales Bookings and

ARR, but that can be solved by having a systematic way to bridge the two numbers. Just make it absolutely clear from leadership down to individual sales reps how the two differ. In cases where you want to drive better alignment, you can create incentives for sellers to pull forward contract start dates.

Discounts

In creating the logic for ARR reporting, it's better to err on the side of conservatism. Reporting a higher number than you might get credit for by investors or the public market is a bad trade in the risk-reward spectrum.

That's why adjusting for discounts in the ARR Build is important. However, not all discounts are created equally. There's nuance in determining which should be removed and which can be ignored.

Let's explore the differences based on the discount window.

RECURRING DISCOUNTS

What to consider: Do you handle discounts applied in perpetuity (forever) on a fixed dollar or percentage-off basis? We see many of these in custom pricing plans offered to early/legacy customers or those with "friends and family" type coupons.

What we recommend: This one's pretty straightforward. You should always remove these from your ARR calculation because they happen on a recurring basis, like the revenue you're annualizing.

FIXED-TERM DISCOUNTS

What to consider: Discounts with a fixed start and end date are more open to interpretation because they're recurring but only for a certain amount of time, e.g., 1, 3, 6, or 12 months. Common sources of these discounts are marketing promotions (e.g., three months free if you use

code "xyz" at checkout), start-up plans (e.g., 50% off or a fixed price plan), or first-year discounts on multi-year Sales contracts.

What we recommend: In the spirit of conservatism, it's better to remove these from your calculation and recognize Expansion when the discount expires. This also ensures you assign the right value between business impact and compensation targets for the company's go-to-market teams.

CREDITS AND ONE-TIME ADJUSTMENTS

What to consider: The last bucket of discounts comes from credits and one-time adjustments. These are one-time and do not relate to the underlying subscription or product bundle. Examples include compensating a customer for a service outage or waiting too long in the customer support queue.

What we recommend: We do not recommend adjusting for these in ARR for two reasons:

- 1. The credits aren't related to the purchase experience or the customer's subscription
- 2. They are not controlled by the teams responsible for revenue (e.g., Sales, Marketing)

Refunds

Mistakes happen. That's what refunds are for.

As an analyst, the problem is that refunds are often used as a catch-all for many things like:

- Incorrect subscription set-up
- A misunderstanding on pricing (e.g., "I didn't know my bill was going to 10x!")
- · Poor customer experiences

They also restate historicals, which can drive you crazy when trying to

reconcile performance data. Rather than walking through each permutation, we'll share our guiding principles on refunds.

What to consider: Always work to fix the underlying issue in the correct upstream source (usually the subscription) and make the best effort to record the refund on the date it was processed instead of applying it retroactively (i.e., avoid historical restatement). The one exception to the latter point is if the refund amount is so high that it materially impacts performance or creates misleading results. For example, when a customer expands by \$1M and your total ARR pre-expansion is only \$200K. In those cases, applying it retroactively makes more sense.

Services and one-time charges

Many companies have service-based offerings in their product suite. This includes things like implementation or priority support.

Similar to discounts, the nature and duration of the service impact whether this revenue category should be included in ARR.

Non-recurring charges

What to consider: Fixed charges related to getting the customer onboarded through things like implementation and training.

What we recommend: We suggest not including these since they are non-recurring in nature. It also avoids a false signal of a Contraction once the implementation period ends.

RECURRING SERVICE CHARGES

What to consider: Other service-based charges that persist through the customer lifecycle, such as priority support.

What we recommend: It's safe to include this in ARR, but we strongly recommend having a way to differentiate between service-based and product-based ARR. This will help you understand what value is

assigned to each component and decide where to invest resources in the business. It's also likely that investors will apply different valuation multiples to product vs. service-based ARR if you go through a fundraising process. It's better to know that upfront and be prepared.

Overages

Overages are usage in excess of the customer's contract. They can apply to seat or usage-based pricing models. A quick example of overages might be if a customer signed a contract for 10 seats but currently has 12 seats in use.

Tracking overages is important because it gives you a sense of how much of your ARR is fixed (i.e., under contract) vs. variable. You may also charge different rates for usage within the contracted amount (through discounts) vs. in excess (through premiums).

There's a quick, two-question test to see if this applies to your business:

- 1. First, do you offer annual contracts?
- 2. Second, can customers increase their usage beyond the contracted amount without signing an updated contract?

If you answered yes to both questions, you should read the following considerations carefully to determine how you want to handle overages in your ARR reporting.

DURATION OF OVERAGES

What to consider: How long do customers typically remain in an overage state once they are incurred?

What we recommend: Erring on the side of caution suggests not including overages, but the duration can help guide your decision if you're on the fence. If customers stay in an overage state past their renewal period, you can probably get away with including them. The effect here is just recognizing the Expansion earlier.

FREQUENCY OF OVERAGES

What to consider: Independent of duration, is it more common for customers to move in and out of overages, or do they tend to stay there once they're in that state?

What we recommend: If customers tend to move in and out of overages, it's safer not to include that in your ARR calculation. Instead, you might want to focus on incentivizing your Sales team to bring those overages under contract before the customer's usage comes back down.

OVERAGE CONCENTRATION

What to consider: How many of your customers are in an overage state, and what does the overage spend represent in terms of your total ARR?

What we recommend: If most of your customers are in an overage state and tend to stay in that state for a long time, you're probably safe to include this in your ARR calculation. However, we advise you to frequently monitor usage patterns across your customer base, as this variable revenue can expose your business, especially in a changing economic environment.

Subscription vs. Invoice view

The final consideration in defining ARR is determining what object to model your reporting off—invoices or subscriptions. Let's walk through both models' pros and cons to determine which is right for you.

Subscription-based view

Subscriptions are generally easier to work with because they have fewer edge cases than invoices. They're also a better proxy for when a customer becomes active (i.e., subscription created) or inactive (i.e., subscription cancelled).

The biggest challenge to working with this object is ensuring you have

the right historical event data to know exactly when and how changes were made to a subscription. Depending on your billing software, this may require the use of webhooks or snapshots of a data warehouse. Once you've established a process to track historical event data, you can accurately model a customer's spending at any given time.

Invoice-based view

Because accessing historical event data on subscriptions can be challenging, many companies model ARR reporting off invoices in the early days.

Invoices provide a full historical archive of customer charges. Every invoice has a Start and End date, making it easy to create a time series view of how spend has evolved over time at the customer level.

While the time series view is easier to produce using the invoice-based method, it does come with a few drawbacks. Let's walk through some of the most common cases below.

TIMING

Not surprisingly, using the invoice-based method requires sending an invoice to start tracking ARR. However, it's certainly not uncommon for there to be a delay between creating or updating a subscription and sending an invoice to a customer. Addressing this delay in invoicing after the fact may require you to back-date invoices, which can cause your historicals to restate.

INVOICE DELAYS

What to consider: How do you avoid gaps in reporting when invoices are delayed or haven't been issued?

What we recommend: First, automate invoicing as much as you can. Second, if manual invoices need to be sent, create SLAs with your billing team on how quickly invoices will be issued after an event. This is

a critical step in locking month-end or quarter-end reporting results. Be sure to communicate these timelines to other teams relying on ARR reporting and instruct them not to pull any ARR reporting until invoicing is complete.

FREQUENCY

Most businesses have customers at different payment frequencies: monthly, quarterly, or annually.

What to consider: How do you compare spend on an apples-to-apples basis when customers are invoiced on different schedules?

What we recommend: Adjust every invoice down to the monthly level and annualize from there. This looks something like the below:

OVERLAPPING INVOICES

In most cases, only one invoice per subscription at a time should be honored to avoid double counting.

CORRECTIVE AND BACKDATED INVOICES

In most cases, only one invoice per subscription at a time should be honored to avoid double counting.

What to consider: How do you avoid double counting when a corrective invoice was issued to resolve a mistake in a prior invoice?

What we recommend: In cases where multiple invoices were issued on the same day, it's usually safe to honor the last invoice created on that date. In cases where a backdated invoice was issued, we typically recommend counting the corrective invoice on its issue date (rather than its backdated start date) to avoid restating historicals. Hopefully, the backdated invoice volume will be low, allowing you to review each instance with your billing team to make a final call. If you find this is a more regular issue, an upstream operational change will likely be required to reduce the volume.

Here's an example of a corrective invoice fix:

Step 1: Create the field

```
ROW_NUMBER() OVER (PARTITION BY customer_id, period_start_date ORDER BY invoice_created DESC) AS invoice_number_by_date
```

- This creates a ranker by customer by period_start_date and sorts descending based on when the invoice was created
- By doing this, the LAST invoice created among matching period_start_dates will return 1

Step 2: Apply a filter

• Filter for invoice_number_by_date = 1 in your data pulls

Here's an example of a backdated logic fix:

```
CASE WHEN invoice_created > invoice_period_start THEN
invoice_created END AS period_start_clean
```

AMENDMENT INVOICES

The second common cause of invoice overlap is from amendment invoices. These come as a result of a customer's subscription changing midway through the service period.

Here's an example:

- Customer A is billed monthly, has five seats on their subscription, and is invoiced on April 1st for \$100 (\$20 per seat per month).
- · On April 16th, they upgrade their subscription to include two more seats.
- Rather than waiting to invoice Customer A for seven seats on May 1st, you issue a \$20 prorated invoice (\$20 * 2 seats * 50% of the month) for the 15 days the two seats are active in April, which were not included in the April 1st invoice.

What to consider: How do you avoid double counting when a prorated invoice was issued to resolve a mistake in a prior invoice?

What we recommend: You should send prorated invoices automatically to bridge the gap between the amendment and the next scheduled invoice. This is the most optimal solution for reporting and cash flow for your business.

There are two things we suggest doing when sending prorated invoices:

- 1. Adjust the end date of the original invoice up to the day before the prorated invoice (i.e., to April 15th in the example).
- 2. Gross up the invoice amount to what it would've been in a non-prorated environment. This will look something like the below:

Now that we've covered all the factors to consider when defining ARR for your business, we can move on to the fun part—building it.

CHAPTER 5

Building ARR

BY CHRIS BURGNER, HEAD OF FINANCE AND ANALYTICS

Bobby's original script for building out ARR at Intercom was thousands of lines of Python.

Here's how that script calculated expansion. It would sort customers' charges by billing number, calculate the difference between the periods, and return the delta or the expansion amount.

```
charges['expansionMRR'] = 0
charges['billingNum'] = 0
for x in set(charges['stripe_customer_id']): #set(charges['stripe_customer_id'])
  temp = charges[charges['stripe_customer_id'] == x]
  temp = temp.sort(['created_at'], ascending = [1])
  for y in range(len(temp)):
        charges.ix[temp.index[y], 'billingNum'] = (y + 1)
        if y != 0:
            charges.ix[temp.index[y], 'expansionMRR'] = (temp.ix[temp.index[y], 'mrr_amount'] - temp.ix[temp.index[y-1], 'mrr_amount'])
expansionMRR = charges['unixMonth', 'expansionMRR']].groupby(['unixMonth']).sum().reset_index()
expansionMRR = expansionMRR.sort(['unixMonth'])
```

Figure 5.1 - Calculating each component of ARR can quickly become difficult to maintain

This ultimately turned into a multi-thousand-line Python script

that pieced various pricing models together. It accounted—pretty horrifically and manually—for all the ways in which somebody could pay us (monthly, quarterly, annually) using all the different methods.

In some ways, it was a pure miracle that this worked, and we were able to keep it all together. The script was brittle, though. It'd break anytime we changed something about our business. Worst of all, nobody could use it but Bobby, meaning nobody else could do ARR analysis—which became a major problem as we scaled.

The most intimidating part was not knowing how to turn this Python script and its output into something that would scale. For Bobby, the Finance team, and everyone else that would ultimately need to report in some way on ARR.

We ultimately settled on the following format—the table we outline in this chapter. It was a complete game changer in how we and the rest of the company accessed ARR reporting.

Now it's time to turn all the logic you've defined into a source of truth. This exercise aims to create a set of tables that you and your team can use to report on ARR easily and efficiently. All of your logic will be applied in various SQL queries.

As you get stuck into this, here are three principles to remember:

- 1. You are ultimately building towards a one-stop-shop reporting table. This table should be comprehensive enough to quickly answer +90% of ARR questions. Be thoughtful about what fields that output table should include to streamline reporting pulls.
- 2. The queries in this exercise will be complex. Be intentional about making them comprehensible. Stay organized in how you write your query. Drop in comments. Modularize the query into as many bitesize components as possible. The goal is to make this query

approachable to others. There's no way to avoid it—this query will be long, but that doesn't mean it should be impossible for others to understand.

3. Your ending ARR table should play nicely with other tables you build around it. This ARR table will help you answer the large majority of questions related to ARR, but it's also the centerpiece for other questions you'll want to ask about the business. That's why you'll want to create a design framework that makes it simple to join additional datasets to the ARR Build.

First, we'll outline the process and pieces you'll put together as part of this exercise. If this isn't quite obvious or intuitive at first, don't worry. We'll walk through each table in more detail and show examples to help illustrate.

- 1. It starts with your Base ARR table. Think of this as creating the ingredients and all the raw data from which you are going to transform your data.
- 2. Within your query, you'll apply the considerations and business logic outlined in the previous table. For example, here, you'll define start and end dates, trial periods, etc.
- 3. That creates a Clean ARR table, which now has all of your considerations and business logic applied.
- 4. Next, you'll create a simple Date table. This table outputs every day of every year across which you have ARR data.
- 5. Now, join your Clean ARR table to your Date table, which creates what we call a Padded ARR table. In other words, it creates a table that shows your Cleaned ARR table with all its business logic applied by customer, for every day.
- 6. From this Padded ARR table, you'll calculate your Net New ARR—or the net differences for each customer daily. You'll use those net calculations to categorize ARR into its component parts: Gross New, Expansion, Contraction, and Churn.

7. From here, you create your Final ARR table. This is the end state. It serves as your source of truth—housing by customer, ARR in its component parts, for every day.

Here's how it looks at a surface level:

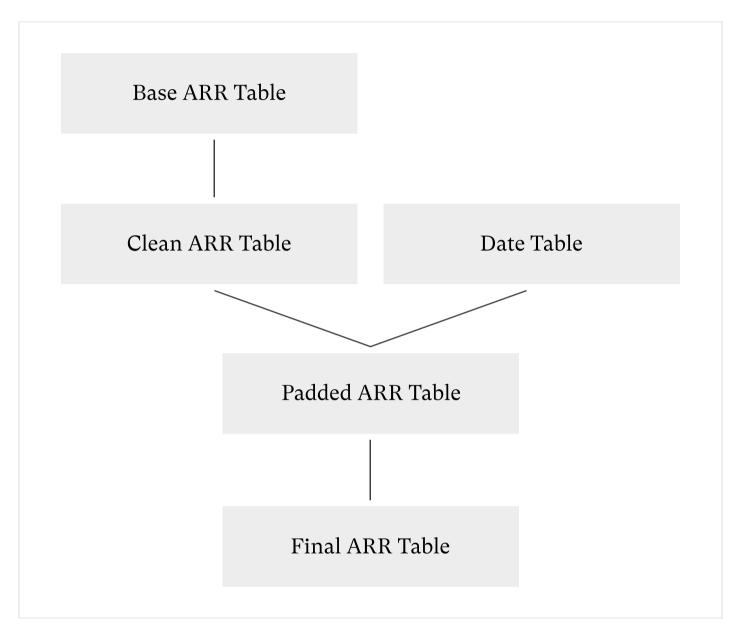


Figure 5.2 - Constructing an easy to query ARR table is a multi-step process

Let's dig into each table in more detail.

The Base ARR Table

This is the query's building block. It's where you'll join across various other tables to bring in the comprehensive set of fields needed to power your ARR definitions.

Note that no calculations are made at this stage. Instead, think of the Base ARR table as a flat file with the raw material to power analysis later on in the query.

Here are some examples of the types of data you'll want to include.

DATE FIELDS

The start and end dates associated with each primary object across the customer base.

PRODUCT AND PRICE MAKE-UP

The IDs and names of product SKUs associated with the primary object and the frequency at which those products are billed (monthly, quarterly, annually, etc.). This detail also allows you to differentiate between service-based (e.g., implementation) and product-based revenue.

CUSTOMER AND TIMING FIELDS

Make sure your logic for defining a customer is included in this table. For this example, we're assuming this happens based on the customer_id.

If you've decided to limit the dataset to only customers who have successfully paid an invoice, you'll also want to bring payment data to this table.

Here's an example of what the output should look like:



Figure 5.3 - The Base ARR table is the building block of all the other tables

The Clean ARR Table

This will get you from the raw Base ARR table to a clean view of the primary object data. The query can get lengthy here, especially with the

invoice-based method, as there are more edge cases to address in the data.

Every company will have unique nuances to contend with, but accounting for the below will get you ~95% of the way there.

ACCOUNTING FOR BILLING FREQUENCY (INVOICE OR SUBSCRIPTION)

As discussed earlier, you'll want to derive the ARR amounts directly in the query by taking the product SKU multiplied by quantity. If you're working with invoices, this will require an extra step of adjusting for invoice duration.

OVERLAPPING AND BACKDATED INVOICES (INVOICE SPECIFIC)

Overlapping invoices can cause double counting as you typically only want to honor a single invoice on any given day for each subscription. Here's how to account for the two edge cases discussed in the previous chapter, Defining Your ARR:

- Corrective invoices (same day): A quick window function can be used to honor the last invoice created on any given day. It's a good idea to review a handful of examples to ensure this logic is correct and to understand better why corrective invoices are being issued.
- Backdated invoices: To avoid restating historicals, create a "start_date_clean" field that defaults to the created date when the created date > the start date on the invoice. As with corrective invoices, review a handful of these examples to ensure the ARR logic matches your current business logic.

• Amendments: The logic used to account for billing frequency can also be applied to calculating ARR values for amendments. The only additional piece needed is adjusting the "end date" of the original invoice (pre-amendment) to one day prior to the amended invoice's issue date.

REMOVING NON-RECURRING ITEMS (INVOICE OR SUBSCRIPTION)

Clean up the dataset for non-recurring products or services and test accounts.

- **Non-recurring items:** Identify any products that are one-time in nature and remove them from your calculations. In your ARR Build, you only want to focus on recurring items.
- **Discounts:** Remove recurring fixed dollar or percent-off discounts from the derived amounts. To enable discount reporting, it may be helpful to surface both gross and net amount fields in your calculations. This can be enlightening from a macro perspective but also useful in enforcing policies on a customer-by-customer basis.
- **Test accounts:** Lastly, always be sure to remove any test accounts from the data. Ideally you have a way to remove these with some kind of 100% off coupon, but keep a close eye out for any emails from your company's domain or customer names matching your employees.

Ultimately, you want to aggregate the data into a single record per primary object id. Here's an example of what the output should look like:

PERIOD_START_DATE_CLEAN	PERIOD_END_DATE_CLEAN	CUSTOMER_ID	INVOICE_ID	SUBSCRIPTION_ID	INVOICE_NUMBER	INVOICE_NUMBER_INV	BILLING_DURATION	ANNUALIZED_AMOUNT
01/30/2023	02/27/2023	1	abc	def	1	12	month	\$1,790.00
02/28/2023	03/29/2023	1	abc	def	2	11	month	\$3,590.00
03/30/2023	04/29/2023	1	abc	def	3	10	month	\$3,590.00
04/30/2023	05/29/2023	1	abc	def	4	9	month	\$3,590.00
05/30/2023	06/29/2023	1	abc	def	5	8	month	\$3,590.0
06/30/2023	07/29/2023	1	abc	def	6	7	month	\$5,390.0
07/30/2023	08/29/2023	1	abc	def	7	6	month	\$5,390.0
08/30/2023	09/29/2023	1	abc	def	8	5	month	\$5,390.0
09/30/2023	10/29/2023	1	abc	def	9	4	month	\$5,390.0
10/30/2023	11/29/2023	1	abc	def	10	3	month	\$5,390.0
11/30/2023	12/13/2023	1	abc	def	11	2	month	\$7,190.0
12/14/2023	12/14/2024	1	abc	def	12	1	year	\$15,273.8

Figure 5.4 - The Clean ARR table should have one row per primary object

The Date table

The Clean ARR table returns a single record for each invoice or subscription object, depending on which you use to generate your build. However, the end goal of the ARR Build is to pull ARR across the entire customer base at any point in time, not just when an invoice or a subscription was created. In order to make this possible, the table needs to expand from a single object record to a record for every day that object was active.

This process is called padding and requires creating a Date table in SQL that the Clean ARR table can join to. Most data warehouses have functions that allow this. Adding a few other dimensions to the date table, such as month_ending or month_beginning fields, is helpful in making reporting easier.

Here's an example of how we build these in Snowflake and the corresponding output:

DATE	WEEK	MONTH	WEEK_BEGINNING	WEEK_ENDING	MONTH_BEGINNING	MONTH_ENDING
12/31/2020	12/28/2020	12/01/2020	FALSE	FALSE	FALSE	TRUE
01/01/2021	12/28/2020	01/01/2021	FALSE	FALSE	TRUE	FALSE
01/02/2021	12/28/2020	01/01/2021	FALSE	FALSE	FALSE	FALSE
01/03/2021	12/28/2020	01/01/2021	FALSE	TRUE	FALSE	FALSE
01/04/2021	01/04/2021	01/01/2021	TRUE	FALSE	FALSE	FALSE

Figure 5.5 — The Date table joined with the ARR tables lets you pull ARR for any date

The Padded ARR table

Next, join the Date and Clean ARR tables together where the dates from the Date table are between the primary object's start and end dates. We call this the Padded ARR table.

Here's what the logic looks like:

```
FROM date_table d

LEFT JOIN clean_arr_table c ON d.date BETWEEN c.period_start_date_clean

AND c.period_end_date_clean
```

The end result returns a record each date the primary object was active. Extrapolating this across all customers makes it possible to pull ARR on any date. (See Figure 5.6.)

TE	WEEK	MONTH	WEEK_BEGINNING	WEEK_ENDING	MONTH_BEGINNIN	MONTH_ENDING	PERIOD_START_DATE_CLEAN	PERIOD_END_DATE_CLEAN	CUSTOMER_ID	INVOICE_ID	SUBSCRIPTION_ID	INVOICE_NUMBER	INVOICE_NUMBER_INV	ANNUALIZED_AMO
01/30/2023	01/30/2023	01/01/2023	TRUE	FALSE	FALSE	FALSE	01/30/2023	02/27/2023	1	abc	def	1	12	\$1,7
01/31/2023	01/30/2023	01/01/2023	FALSE	FALSE	FALSE	TRUE	01/30/2023	02/27/2023	1	abc	def	1	12	\$1,7
02/01/2023	01/30/2023	02/01/2023	FALSE	FALSE	TRUE	FALSE	01/30/2023	02/27/2023	1	abc	def	1	12	\$1,7
02/02/2023	01/30/2023	02/01/2023	FALSE	FALSE	FALSE	FALSE	01/30/2023	02/27/2023	1	abc	def	1	12	\$1,7
02/03/2023	01/30/2023	02/01/2023	FALSE	FALSE	FALSE	FALSE	01/30/2023	02/27/2023	1	abc	def	1	12	\$1,7
02/04/2023	01/30/2023	02/01/2023	FALSE	FALSE	FALSE	FALSE	01/30/2023	02/27/2023	1	abc	def	1	12	\$1,7
02/05/2023	01/30/2023	02/01/2023	FALSE	TRUE	FALSE	FALSE	01/30/2023	02/27/2023	1	abc	def	1	12	\$1,7
02/06/2023	02/06/2023	02/01/2023	TRUE	FALSE	FALSE	FALSE	01/30/2023	02/27/2023	1	abc	def	1	12	\$1,7
02/07/2023	02/06/2023	02/01/2023	FALSE	FALSE	FALSE	FALSE	01/30/2023	02/27/2023	1	abc	def	1	12	\$1,7
02/08/2023	02/06/2023	02/01/2023	FALSE	FALSE	FALSE	FALSE	01/30/2023	02/27/2023	1	abc	def	1	12	\$1,7
02/09/2023	02/06/2023	02/01/2023	FALSE	FALSE	FALSE	FALSE	01/30/2023	02/27/2023	1	abc	def	1	12	\$1,7
02/10/2023	02/06/2023	02/01/2023	FALSE	FALSE	FALSE	FALSE	01/30/2023	02/27/2023	1	abc	def	1	12	\$1,7
02/11/2023	02/06/2023	02/01/2023	FALSE	FALSE	FALSE	FALSE	01/30/2023	02/27/2023	1	abc	def	1	12	\$1,7
02/12/2023	02/06/2023	02/01/2023	FALSE	TRUE	FALSE	FALSE	01/30/2023	02/27/2023	1	abc	def	1	12	\$1,7
02/13/2023	02/13/2023	02/01/2023	TRUE	FALSE	FALSE	FALSE	01/30/2023	02/27/2023	1	abc	def	1	12	\$1,7

Figure 5.6 - The Padded ARR table now reports ARR on each day

The Final ARR table

With the Padded ARR table in place, you have the foundation to calculate Net New ARR. This is derived by taking the ARR on any given date and subtracting it from the ARR on the prior date. The easiest way to derive this is through the LAG function in SQL:

```
annualized_amount - COALESCE(LAG(annualized_amount,1) OVER
(PARTITION BY customer_id ORDER BY date ASC),0) as
net_new_arr
```

Once Net New ARR is calculated, you can start dropping these values into each ARR component part's calculation depending on:

- 1. Where you've drawn the line for counting a customer
- 2. The state of the customer at the time of Net New ARR

Here are some examples of how to think about this:

DATA OUTPUT

The Final ARR table brings everything together on a per-customer, perday basis and can easily be aggregated from there. Here's how it should look:

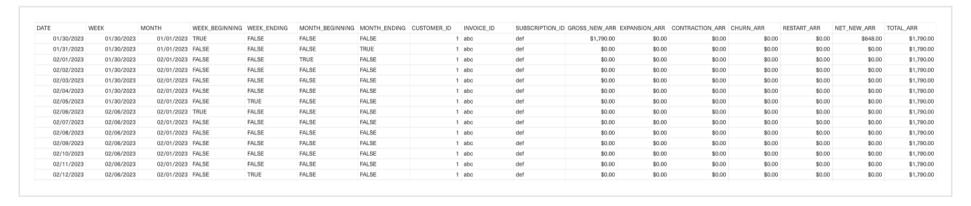


Figure 5.7 - The Final ARR table breaks the ARR changes into its component parts

INCORPORATING ADDITIONAL DATASETS

With the ARR Build complete, you may consider bringing in other datasets to join against the Final ARR table for other reporting needs, such as product usage, customer acquisition, and firmographic data.

The most efficient way to do this is to establish separate, modular groupings of data and eventually join those together in one large "Comprehensive" ARR table. Under this structure, modular data sets are grouped together when they share a similar classification or upstream data source. (See Figure 5.8.)

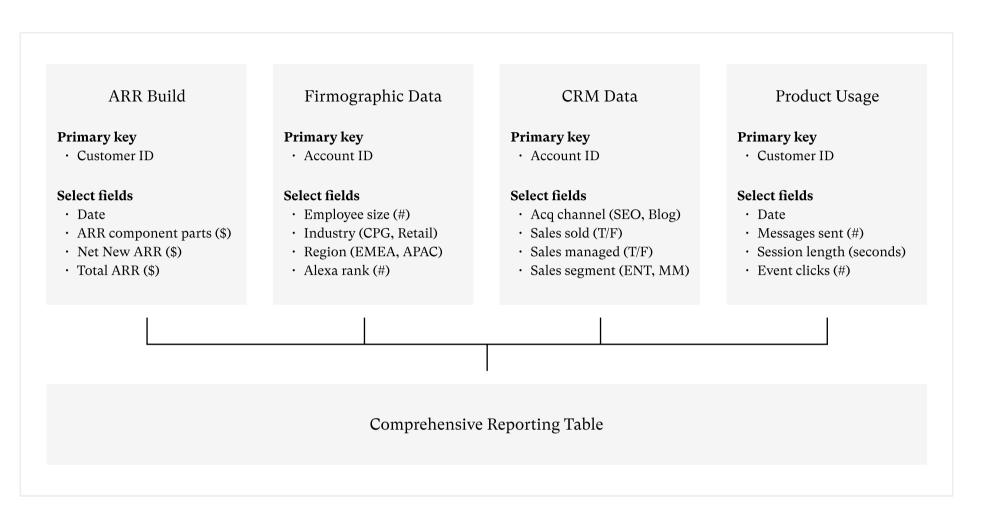


Figure 5.8 - Further simplify reporting by enriching ARR data with other data about your business

There are a few advantages to organizing your data like this:

- 1. You create a one-stop-shop table. This single table can handle ~90% of your reporting, from board decks to go-to-market updates.
- 2. You simplify the SQL writing. Creating modular datasets creates a single source of truth, rather than rewriting it each time you want to query the data. Also, updating logic happens in one location and automatically flows through to your downstream sources.
- 3. You gain a lot of efficiency in debugging and comprehension. The rest of the team can get up to speed on your ARR table more quickly, introducing redundancy and better data comprehension.

Here's every field we ended up with in our table at Intercom:

ARR TIME SERIES	
Gross New	A unique identifier for each ARR instance that is easily linked to other data sets
Day	Run every column on this table for every accound_id daily
Customer Start Date	As defined in your business logic, the day this customer became a paying customer
Customer Cancel Date	As defined in business logic, the day this customer churned
Price plan/product name	Some indication for the plan or product this customer has on that day
Contract type	e.g. Multi-year, annual, semi-annual, quarterly, monthly
Payment terms	e.g. Multi-year upfront, annual upfront, semi-annual, quarterly, monthly
Most recent invoice amount	Amount paid on last invoice
ARR amount	ARR amount as defined by business logic (e.g. taking into consideration discounts, ramping, usage, overages, underages, etc.)
Gross New ARR	Gross new ARR amount relative to day -1
Expansion ARR	Expansion ARR amount relative to day -1
Contraction ARR	Contraction ARR amount relative to day -1
Churned ARR	Churned ARR amount relative to day -1
Net New ARR	Net New ARR amount relative to day -1
Month start date	Start date for the month
Month end date	End date for the month
Month beginning	True / False
Month ending	True / False
Quarter start date	Start date for the quarter

Quarter end date	End date for the quarter
Quarter beginning	True / False
Quarter ending	True / False
Fiscal year start date	Start date for the fiscal year
Fiscal year end date	End date for the fiscal year
Fiscal year beginning	True / False
Fiscal year ending	True / False

Figure 5.9 - Intercom's ARR table had many convenience columns to simplify reporting

CHAPTER 6

Acting on ARR

BY BOBBY PINERO, CO-FOUNDER AND CEO

"Looking at ARR every day was one of the most impactful things we did at Intercom. The Daily Pulse - literally, that's what we called it - accelerated our ability to make decisions and understand our evolving business."

Eoghan McCabe, CEO and Co-Founder of Intercom

One simple report changed the trajectory of Intercom. Interestingly, there were no insights or opinions to be found in the report. The Daily Pulse was an auto-generated email with simple graphs sent out to everyone in the company every day at 5 pm PST, summarizing our business that day. To say the metrics were simple would be an overstatement. It's about as basic as reporting gets, mostly centered around ARR.

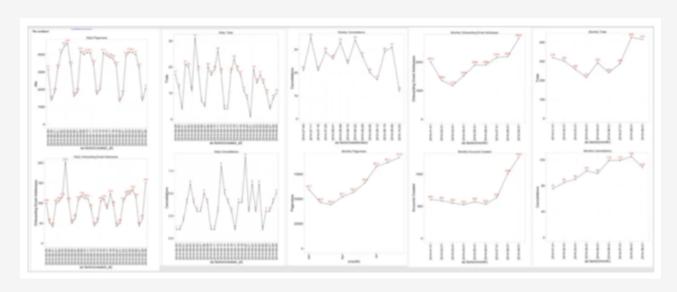


Figure 6.1 - Intercom's KPIs were auto-emailed to everyone in the company, every day

The charts were: Daily Page Views, Daily Onboarding Emails, Daily App Installs, Daily Trials, Daily Cancellations, Daily Gross New, Expansion, Contraction, and Churned ARR-including weekly and monthly versions of each of those charts.

Ask any early Intercom employee and they'll remember its impact. It did three things:

- 1. Pushing metrics to every employee daily allowed everyone to own what we cared most about at the time growth which pervaded decision-making across the entire company.
- 2. It forced a check in vevery vingle vay. You better believe that if something looked funny at the end of the day there was a reply-all from our CEO, Eoghan, or one of our function heads. We were all on the hook to chase down answers within 24 hours, which drove extreme accountability.
- 3. We had incredibly fast feedback loops. Launch something on Product Hunt? Introduce a new screen in onboarding? Break some instrumentation in the funnel? You would see early indicators of the impact the same or the next day.

This fundamentally changed the culture of the company and how people engaged with data.

The Daily Pulse is an example of how operationalizing metrics matters almost more than the metrics themselves. We had <u>ARR reporting</u> set up before that, but it wasn't until we started pushing it to people daily that folks understood it, built an intuition about it, and could connect the dots between their work and how it moved (or did not move) the needle.

Most finance folks spend an inordinate amount of time setting up the reporting and getting things correct but don't spend enough time thinking through how to let others make the most of that information. In this chapter, we share a few principles to help with that.

Build a fast and consistent cadence for looking at metrics

Most of us organize our finances and pay bills monthly. Companies have board meetings and report quarterly. These are easy cycles to fall back to —even small startups that compete on speed fall back to reporting monthly or quarterly. A surefire way to create complacency around data is to fall back to these cadences.

Decisions are made daily—often multiple times a day, particularly in startups. You need a set of metrics that match the cadence at which you're making decisions. As was the case with the Daily Pulse, we've found that looking at ARR metrics on a daily basis actually makes a difference.

We've also found that having a weekly sync where you pull up on the high-level metrics across the business—which should include ARR, its component parts, a forecast, and cohorts—keeps everyone on the same page. Every week on Friday, we'd walk the entire exec team through the following:

- 1. Every metric in our funnel
- 2. The status of every ongoing experiment
- 3. Our ARR Build, its parts, and a forecast

We did this every single Friday. It was a nightmare to put together—it'd

take me all day to do—but it was incredibly impactful. Here's a sample from one of the decks back then:



Figure 6.2 - Reviewing the entire business once a week keeps everyone on the same page

Deliver metrics where your team works

We build dashboards because we assume people will log in regularly to view them. But they don't—people are lazy, busy, or both. No matter how important the dashboard or metric is, days are quickly consumed by meetings and never-ending to-do lists. Teammates will go weeks before logging in to view your beautiful dashboard.

The fastest way to change how data gets used in an organization is to deliver where people are already working, i.e., "The Daily Pulse". Sending key reports and metrics daily via email or chat will quickly change your teams' understanding of and ability to act on data.

Developing habits requires making small changes on a consistent basis. To make data habitually part of the conversation, force it to be part of someone's every day. (See Figure 6.3.)

Keep it stupid simple

If you send your team metrics, you must ensure they understand exactly

what they are looking at. They should be able to articulate why they matter and how they're calculated. Most people think this involves creating a data dictionary or a repository where the team can find the definition for every metric. But if that's your solution the metrics are too complicated to start with. You should pick inherently simple metrics rather than spend time on detailed explanations.

For example, say you want your team to stay on top of Net Revenue Retention (NRR). It's a quite complex metric that's multifaceted and not widely understood. We filled many pages in this book covering it and could probably write another entire book on the topic! Most people in the business don't really understand it. Alternatively, you might consider showing the team the actual number of customers who made it to X period trended over time or the amount of expansion/contraction dollars over time. Those are metrics everyone understands and feels empowered to act against.

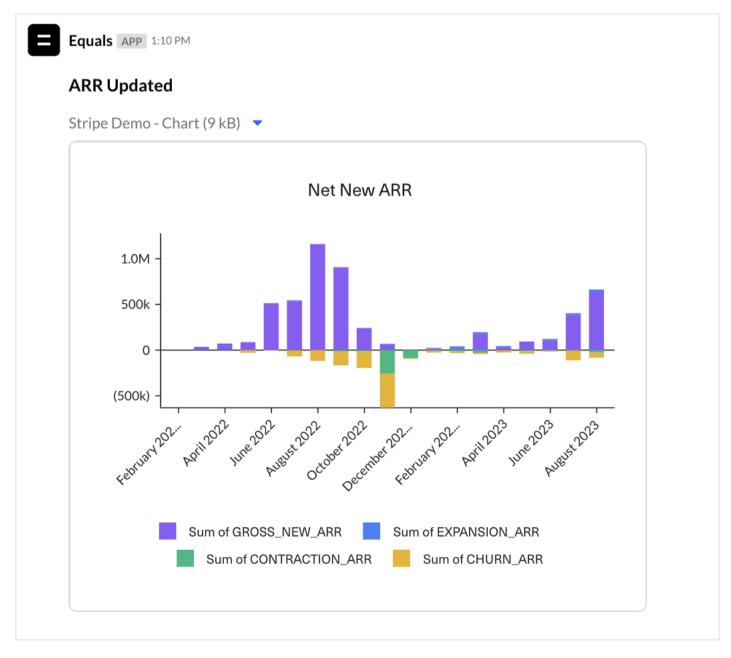


Figure 6.3 - An example of our "Daily Pulse" now at Equals, sent to Slack using Equals

Tie everything back to ARR

This is why we suggest starting with ARR as the centerpiece for your reporting. Every business question ultimately points back to the same thing: Is this a long-term profitable thing for us to do? Is there a measurable return on investment on some time horizon? The best way to make sure that every decision you make—and that every decision others make—meets those criteria is to tie as many things back to ARR as possible.

For example, if you're trying to establish a user engagement metric, you don't want to create that metric in a vacuum, devoid of any consideration for the impact it might have on the broader business. You want to create that metric by determining what type or level of engagement correlates well with a customer who's either more likely to pay you (Gross New), pay you more (Expansion), or pay you for longer (Retention). From there, you've established a connection between that engagement metric and why it matters to you (ARR).

Play great defense

We'll share another story that one of our early mentors shared. Ray Ko was the head of Analytics and Growth at Facebook in the early days. He was in charge of understanding Facebook's core metric—MAU (Monthly Active Users)—and why it grew. He built the systems that supported it, and since his Facebook days, he's gone on to advise many other top Growth teams in the valley. He has a thesis that growth is actually unlocked by playing great defense in your business. What does that mean? Well, this is his story.

Early in the Facebook days (before they even had mobile!), they were clearly focused on growing the number of monthly active users on the platform. At one point, though, they started to see a meaningful slowdown in MAU. As they broke it down into its component parts (new, churned, retained), they saw that the

number of new active users was the cause. They broke that down further and investigated all sorts of different cross-cuts. Was it certain geographies? Different demographics? Had they introduced a bug, etc.? The investigation actually found that one of their email providers was throttling confirmation emails.

When someone signed up for Facebook, they'd click join, but then they'd have to go back to their inbox to click through a confirmation email to confirm it was actually them. Yahoo was apparently throttling those emails, which was the cause of lower new signups. Ok, that makes sense. They had the systems in place to detect that problem. And the easy answer from there was to work with Yahoo! to prevent any throttling moving forward.

However, the real insight came from seeing that confirmation emails were a critical step in the signup process. Previously, that step was taken for granted. But from here, the opportunity to improve actually became quite clear—and it was revealed by something being broken. So, the team started to ask a bunch of different questions that they had never considered previously. How can we make confirmation emails go out faster? How can we make them clearer? Do we even need them in the first place? All of which was in the spirit of being able to grow faster.

We love this lesson because it speaks to the importance of setting up a great system. That system is what lets you play great defense. You can play great defense by building a system that gives you clear visibility into your ARR and business processes—consistently and at scale. The question then becomes how can you take that defense and turn it into great offense? What's the equivalent in your business for confirmation emails, and how can you not just fix the problem but turn it into an opportunity to grow?

CHAPTER 7

Benchmarking ARR

BY BOBBY PINERO, CO-FOUNDER AND CEO

Truth be told, I hate benchmarks.

In the early Intercom days, we would always get dinged for our churn. We'd always have to spend time carefully crafting our story on churn when we went out to fundraise. It was the reason many passed. And it constantly came up in board meetings.

Here's the thing, though. Whenever it came up, it was in reference to other businesses. But most other SaaS businesses at the time were very different to Intercom. In 2013, the only notable public SaaS companies were Salesforce and Workday, and most private SaaS companies of any meaningful scale were companies similar to them—enterprise companies. Intercom was different. We were part of a new generation of SaaS companies. One that sold

primarily to other startups. Smaller contracts, paid monthly, and self-serve purchases. That might be commonplace now, but it wasn't then.

That's why benchmarks simply didn't apply to us. It wasn't fair to hold us to retention rates similar to a Salesforce and certainly not a Workday. The cost of acquiring customers for those companies is way higher, so they need to have meaningfully different retention dynamics. Customers came to Intercom essentially for free. We could churn them at a much higher rate and still have a beautiful business. I share this as a cautionary tale before you read this chapter.

Be careful what benchmarks you use and who you compare your business against. Every business is different. A subscription prosumer business is different from a bottoms-up low ACV business, which is different from an enterprise business with a much higher ACV and fewer customers. Even within those categories, businesses can look meaningfully different. And, of course, the stage of business matters, too! A seed-stage business can and will have different growth, retention, and acquisition dynamics from those at scale. Seems obvious and makes sense, but often, benchmarks just get tossed around as a stick to beat you with. Don't let that happen.

Your startup is bringing something new to the world. That's what it means to be a startup. And that newness can mean that existing benchmarks either don't apply or they need to be meaningfully discounted in value.

Let this serve as a gentle reminder that what's far more pragmatic is treating benchmarks as a reference. You need to take the time to deeply understand the dynamics of your business and articulate what's working and what's not. You must define what's a good or bad metric in relation to where you want to go, which will not always be where others have been. Sometimes you just have to chart your own course.

Growth benchmarks

A simple place to start is to consider ARR thresholds and growth rates for different-stage companies. These are by no means hard and fast rules. You can raise venture rounds at meaningfully different benchmarks from this. Other factors that get weighed heavily at different stages include the strength of the team, the complexity of the product, TAM, product engagement and usage, and other metrics. To give you an example, an investor might bet on a product that has incredibly high adoption and usage metrics, but that has not yet matured its go-to-market motion in the belief that high product usage indicates a fantastic fit with users, but the startup has not yet nailed how to sell its product.

	GREAT	GOOD	DANGER ZONE
Series A	\$1M ARR, 3×+ YtY	\$1M ARR, 2–3×+ YtY	<\$1M ARR, <2×+ YtY
Series B	\$5M ARR, 3× YtY	\$5M ARR, 2–3×+ YtY	<\$1M ARR, <2×+ YtY
Series C+	\$20M ARR, 2–3× YtY	\$20M ARR, 1–2×+ YtY	<\$20M ARR, <1×+ YtY

Figure 7.1 - ARR can be benchmarked by absolute amount and growth rate

Retention benchmarks

Retention is one of the trickiest things to benchmark—queue the story from Intercom. It meaningfully depends on the type of recurring revenue business, but also the stage of the company and maturity of the business. It's common that as a business scales, much of its cohorts are more mature, meaning they churn less frequently but also potentially expand less. There are many forces pulling retention in different directions. For example, an early-stage company might have lower retention because they are selling to smaller companies. Or because their offering is less mature. Or because their cohorts are all earlier in their lifecycle.

However, they might also have higher retention because expansion often occurs earlier in a customer's lifecycle.

Finally, retention is a very mixed bag in how it's calculated. To give you a feel for this, Keybanc has a 40-page deck that helps define how each public company calculates retention. They're all different. Here's a snippet from it:

COMPANY	DEFINED TERM	METRIC FORMULA	RATE AT IPO	DEFINITION
APP DYNAMICS⁽¹⁾	Dollar-Based Net Retention Rate	\$ implied recurring ACV LTM end customers LTM – 1 end LTM end customers LTM – 1 end \$ implied recurring ACV LTM-1 end	127%	"Our dollar-based net retention rate compares the recurring contract value from the same set of customers across comparable periods. Given the repeat buying pattern of our customers and the average term of our contracts, we measure this metric over a set of customers who have been with us for at least one full year. To calculate our dollar-based net retention rate for a particular trailing 12-month period, we first establish the recurring contract value for the previous trailing 12-month period. This effectively represents recurring dollars that we should expect in the current trailing 12-month period from the cohort of customers from the previous trailing 12-month period without any expansion or contraction. We subsequently measure the recurring contract value in the current trailing 12-month period. Dollar-based net retention rate is then calculated by dividing the aggregate recurring contract value in the current trailing 12-month period. Recurring contracts are time-based arrangements for subscriptions and do not include perpetual license or professional services arrangements." (S-1/A filed on 1/24/17, Page 14)
athenahealth	Contract renewal rate	\$ implied ACV t end \$ implied ACV customers t-1 end \$ implied ACV customers t-1 end t-1 end	97%	Dollar-based renewal rate including the benefit of upsells, based on ACV "We have experienced a contract renewal rate of at least 97% in each of the last five years." (424B4 filed on 9/19/07, Page 1) "This rate reflects the implied annualized contract value of the customers at period end who were also customers at the end of the prior period, divided by the implied annualized contract value of the customers at the end of the prior period." (athenahealth Investor Relations)
XAtlassian	Net expansion rate	Trailing 4 Qtr. Avg. \$ revenue \(\frac{customers q-4}{q} \) \$ revenue \(\frac{customers q-4}{q-4} \)	>100%	Dollar-based expansion rate including the benefits of upsells, based on GAAP revenue "Such expansion is measured by our average quarterly net expansion rate, which calculates the year-over-year change in quarterly spending by customers that were paying customers during the same quarter in the prior year ("Prior Year Cohorts"), which is net of lost customers or reduced usage within a customer. Our average quarterly net expansion rate has been more than 100% for each quarter during fiscal 2014 and fiscal 2015." (424B4 filed on 12/10/15, Page 63-64)

Figure 7.2 - Measuring retention is tricky and differs business to business

That said, we want to share probably the most helpful resource we've found to help you calibrate whether your retention is Good or Great. Anything worse than that, you're in the Danger Zone again. These benchmarks come from Lenny's Newsletter, and what we like so much about it is that it breaks things down according to the type of SaaS business you run.

	GOOD	GREAT
Consumer SaaS	~55%	~80%
Bottom-up SaaS	~100%	~120%
Land and Expand VSB SaaS	~80%	~100%
Land and Expand SMB / Mid-Market SaaS	~90%	~110%
Enterprise SaaS	~110%	~130%

Figure 7.3 - NRR benchmarks from Lenny's Newsletter

For ARR, those are the benchmarks we trust the most. To reiterate, the most important message we can leave you about benchmarks is to use them wisely. Never let benchmarks drive how you run your business. Always ensure you do that from first-principles thinking and your own understanding of your unique business.

Taking Control

BY BOBBY PINERO, CO-FOUNDER AND CEO

We have a lot of empathy for that first finance hire. Everything outlined in this book is daunting to take on and make it all work. And there's the added pressure of feeling the need to get it right. We know because we've been there ourselves.

If you feel stuck or overwhelmed, please reach out. We're here to help. As a part of Equals, we offer support in building all of this. We've helped many SaaS companies set up their own <u>ARR reporting</u>.

"Getting ARR from 0 to 97% is somewhat easy - getting it from 97 to 100% is extremely hard. That's where the Equals team was super valuable."

— Ibrahim Cisse, VP of Finance at Descript

Want more?

We got you. Here are some of our favorite resources related to ARR and subscription-based business models.

SaaS Metrics 2.0

This is the holy bible of SaaS metrics. It's where I learned the fundamentals of how a recurring revenue business works. It does a nice job of explaining why recurring revenue models are different, why the cash flow dynamics are different, why the recurring nature of revenue matters, and how to think about the lifetime value of said revenue. It's everything. Read it.

Why are SaaS companies valued on revenue multiples?

Every business outside of SaaS is valued on earnings and profit, not revenue. Yet SaaS companies are valued on revenue multiples. Why? Read this to find out.

How Figma measured Self-Serve ARR

This is a deep dive into one of the greatest technology companies of all time and how they measured their business. It covers the exact model they used to measure and forecast self-serve ARR at the time.

How Snowflake forecast ARR

Another one of the great SaaS companies of our generation shows how they built their ARR forecasting process. It's fascinating.

SaaS Metrics for Fundraising

If you're a SaaS business considering raising capital, ARR is one of the core metrics you need to report strongly on. But what else will investors ask you along the process? This handy resource covers everything you need to prepare.

Meritech SaaS Index

In general, we're wary of benchmarking for all the reasons discussed previously. However, Meritech have put together here a great resource that pulls key metrics from every SaaS public company.

Clouded Judgement

This blog is one of the best places to stay up to speed on the most recent SaaS trends and public company earnings.

Thanks for reading!

Bobby and Chris



A book by **Equals**