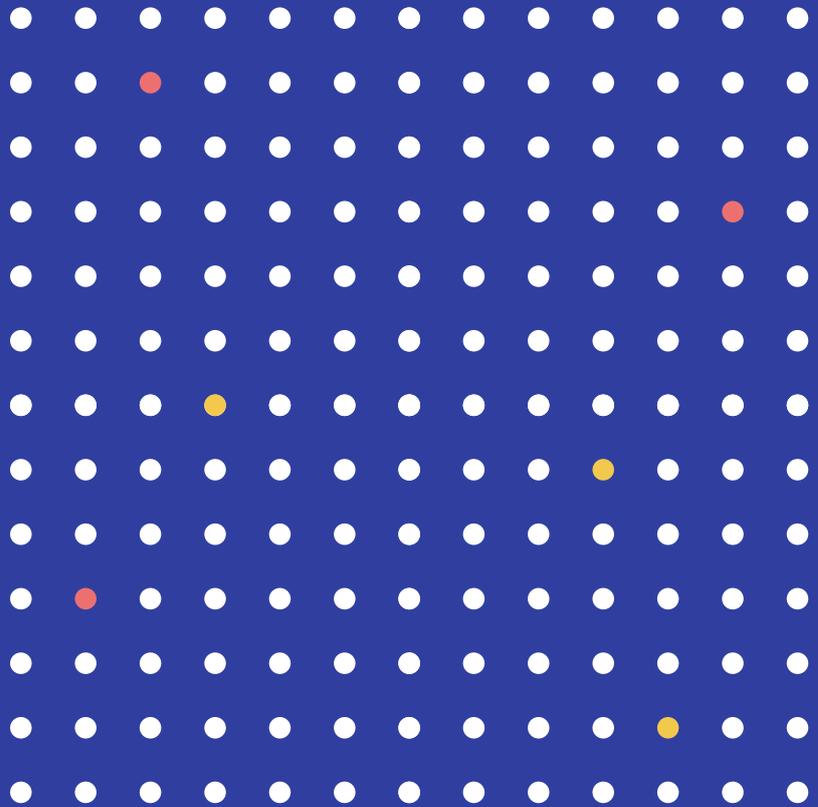
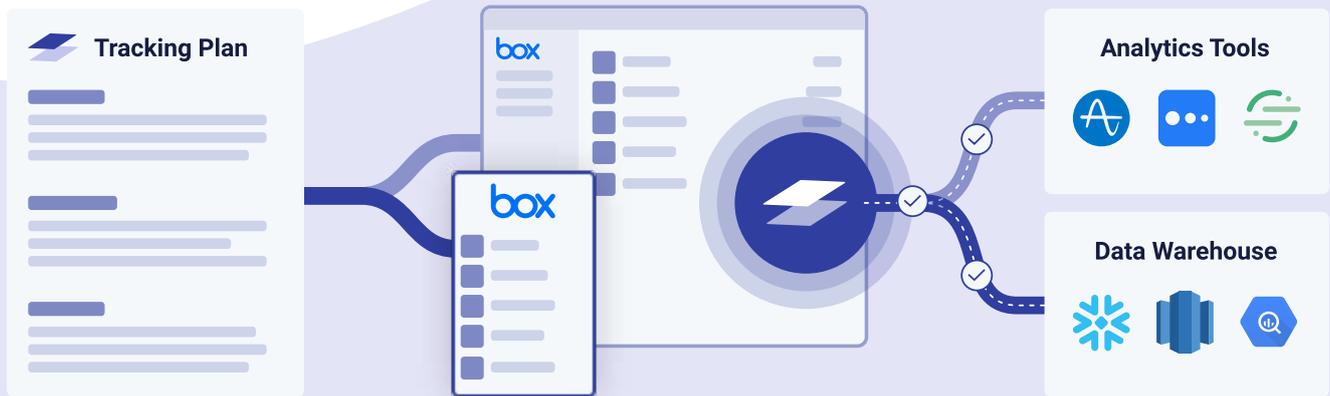


# A comprehensive guide to event tracking

How to take your event tracking to the next level for accurate and complete data you can trust





We've been looking at how organisations use customer data to improve their products for over three years now: we've spoken to over 400 individuals in data and product teams, and what we found is that **most data quality issues stem from inaccurate and incomplete event tracking.**

This journey of problem validation has greatly informed the kinds of features that we've built into Iteratively. It's also given us huge insight into what works and what doesn't when it comes to analytics tracking.

This document offers **practical guidance** on building an event tracking strategy. Whether you're starting from scratch, or improving an existing strategy, we take you through a phased plan on how to approach this.

# What's in this paper?

We've laid out our recommendations in the order that we think they should be done: how to start off this process, and how to keep it going. You may be very aware of some of the recommendations here already, so feel free to skip to what's most relevant to you.

1

## **WHY YOU SHOULD CARE ABOUT EVENT TRACKING**

Data helps you make good decisions, and data professionals spend a surprisingly small amount of time actually extracting insights.

2

## **START WITH YOUR GOALS, NOT YOUR DATA**

Some common misconceptions on the power of data.

3

## **WHAT'S A TRACKING PLAN?**

Important to know, because this paper takes you through how to build one.

4

## **WHO OWNS THE TRACKING PLAN?**

The answer is not 'everyone'.

5

## **ESTABLISHING PROCESSES**

In this section we really tackle the fine details. Defining shared taxonomies; implementation; documentation; and most importantly: enforcement of the plan.

6

## **HOW TO GET STARTED – A CHECKLIST**

# 1. Why you should care about event tracking

No matter the size of your organisation, or the kinds of products you build, one thing remains true: you cannot leave all decision making up to 'gut feelings' or qualitative feedback. Asking questions of your data is the best way to get good, actionable answers that can help you improve your customer experience and ultimately your bottom line.

**You've probably heard the saying: garbage in, garbage out.**

If understanding how users interact with your products is the ultimate way to improve your products, you need to be collecting and analysing data that you can trust. The answers you get out of the data are only as good as the data itself.

## Another important consideration: time spent on cleaning up data



**45%**

The [2020 State of Data Science](#) report by [Anaconda](#) has shown us that data analysts will spend around **45%** of their time cleaning and loading data. Minimising that would mean more time spent doing actual insight work.



**11%**

Another study by [Kaggle](#) revealed that data scientists, on average, only spend about **11%** of their time extracting insights.

The first step to clean and accurate data is to ensure you're collecting it correctly. Data collection is the foundation of your data stack, but it's often overlooked and so many companies don't spend the time and resources needed to get it right.

---

**You're reading this, so we know you care – let's dive in to figure out how you can improve your data collection and data quality.**

---

## 2. Start with your goals, not your data

Data is like Google Docs: nearly everyone uses it in some capacity. That's because it's an extremely powerful tool. However, many organisations mistake data itself as their 'differentiator'. This isn't quite true, because anyone building digital products and services will produce data. Therefore, **your differentiator is how you use your data.**

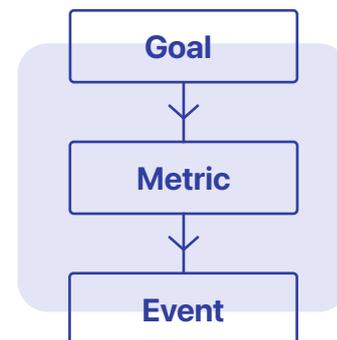
This common misconception about how to use data often leads to developing strategies that use data as the starting point – you would never do this with Google Docs.

When building an analytics strategy – and before you start tracking anything – start by looking at what your goals are, instead of looking at how you might leverage data that you haven't captured yet.

### WE'LL TAKE A SIMPLE EXAMPLE:

maybe you want to increase acquisition by **15% in Q1**. Having some solid ideas on how you're going to do that is great – but before even implementing those ideas, you need to figure out how you're going to measure success. So, ask yourself:

- By what metric(s) can we measure acquisition? Conversion rate could be one
- What event(s) do you need to track in order to know your conversion rate? Things like unique site visits, and user sign ups.



Honing-in on this one goal has shown us that you don't need to track 'everything' in order to answer your questions. Optimise for the key questions and goals first, and then go from there.

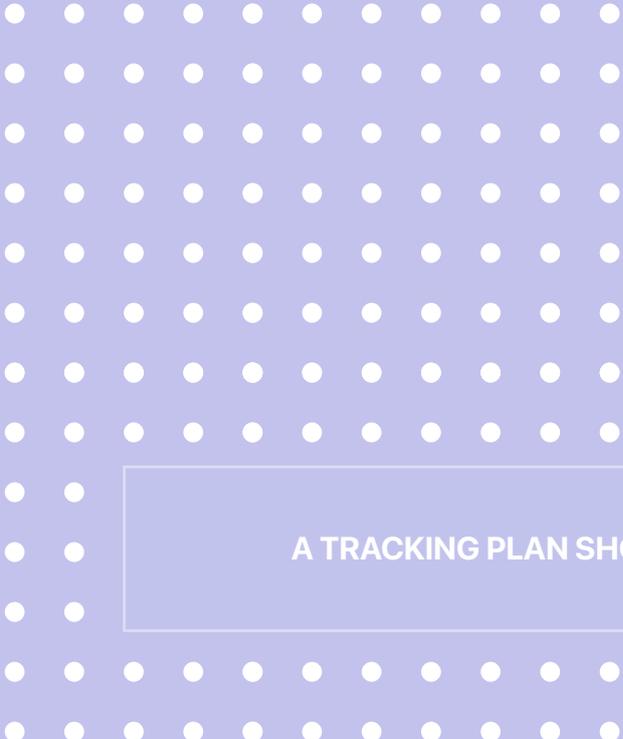
**Making data-driven decisions isn't about how much data you have, or even the tools you use. It's about asking the right questions.**

### 3. What is a tracking plan?

Before you start collecting your data, we strongly recommend that all relevant stakeholders meet to agree on what should be tracked.

A tracking plan helps codify a **single source of truth** for what matters to your organisation. It provides you with structure and can empower you and your teams to standardize data collection across your products. It also helps data consumers understand what data is being captured and what that data means.

Once you've defined your goals, you're ready to put together a tracking plan.



#### A TRACKING PLAN SHOULD SEEK TO ANSWER THESE QUESTIONS

What data is needed to answer those metrics?

What are the metrics we care about as a company?

How can I ensure the data being captured is trustworthy?

## Tracking plan examples

Playing around with live examples of tracking plans can be really helpful – here are few templates you can look at right now. Note that all of these include business goals, to justify what is being tracked.

[Mixpanel](#) has put together a few tracking plans, specific to certain verticals.

[Amplitude](#) has put a highly detailed one together in Google Sheets.

[Segment](#) also has a good example to get you started.

Remember that these are just some examples to get you started – your tracking plan should be unique to your business. And they don't encompass all the best practices that we've outlined in this guide, so do read on.

## A note on implicit and explicit data capture

Before diving into how to put together a great tracking plan, it's useful to consider whether you're leveraging explicit or implicit data capture. There are upsides and downsides to both, and which one you pick can depend on a number of factors. We've outlined the pros and cons of each and you can [discover which is right for you in this blog post](#).

**Implicit event tracking** is where you don't define the events you want to track beforehand; you simply collect all user interactions and decide what's useful later. This is also known as codeless event tracking.

Example tools: [Heap](#), [FullStory](#)

**Explicit event tracking** is where you manually define which events you want to track, using code-based analytics. Using this method means you have an idea of what metrics and interactions are interesting to you, before the tracking even begins.

Example tools: [Amplitude](#), [Iteratively](#)

We **strongly recommend** explicit data capture for anyone serious about leveraging their data for decision-making and the rest of this white paper makes recommendations for teams who rely on explicit data capture.

## 4. Who owns the tracking plan?

Event tracking is an ongoing collaborative process. There are a range of stakeholders across multiple teams: product managers, data analysts, engineers, and the executive will all have some level of investment in the tracking plan. But who should take **ownership** of it?

Many organisations tend to default to the idea that everyone should pitch in and make sure the tracking plan is enforced, and consistent. In practice, this doesn't work. If keeping the tracking plan up to date is considered everyone's responsibility, then no one is truly accountable.

### THE RESULT PRESENTS YOU WITH TWO PROBLEMS:

1. A messy, out of date spreadsheet, and the last remnants of any documented event tracking lost on an unactioned Jira ticket somewhere. No one will take it upon themselves to clean this up, because it's hard to know where to begin. And besides, it was meant to be 'everyone's' job.
2. Missing out on key insights. Sometime after a new feature release, the product team or your CEO will want to know how that feature is performing. Without a dedicated owner, new features often go out without tracking and you'll have no data to answer your important questions or to continuously improve that feature.



**PM works on a release**



**Release happens**



**CEO asks the PM how it's performing**



**PM: Let me ask data team**



**Data team: you never brought us in – there is no data on this feature**



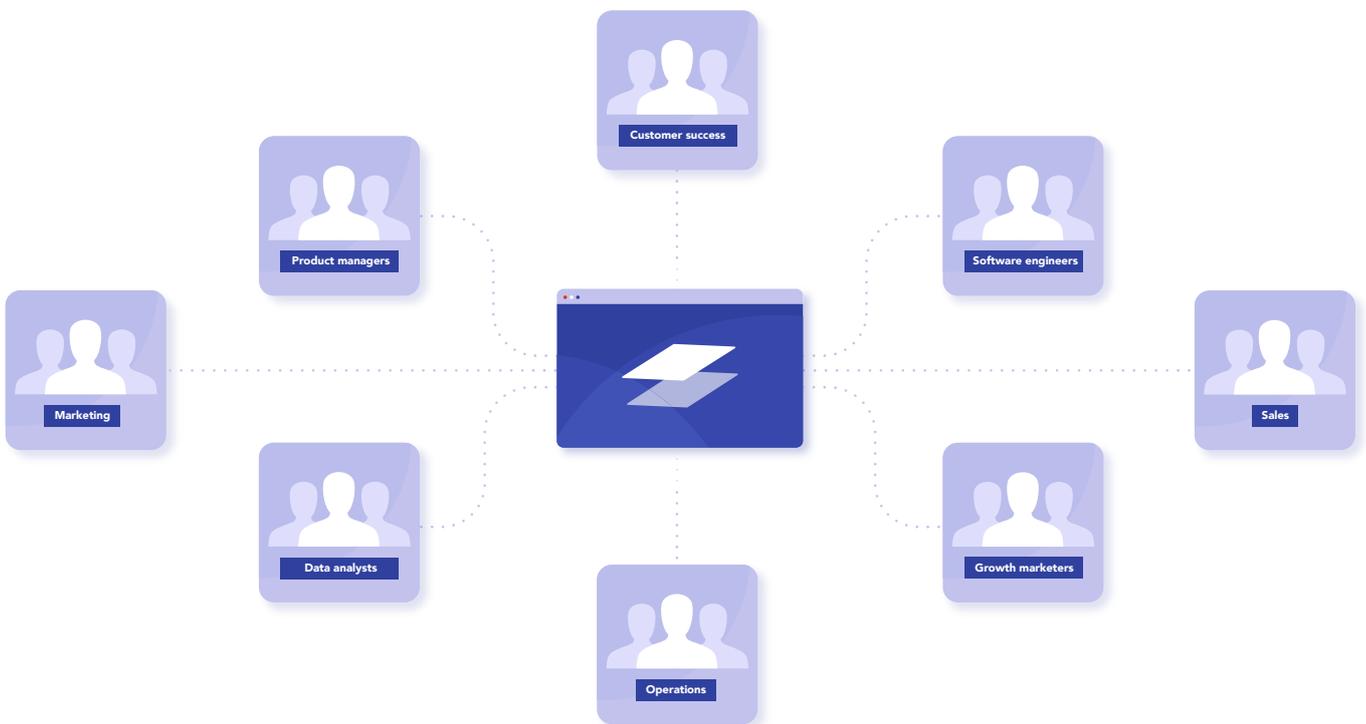
**PM goes back to the CEO with no answers**



**Data team and PM are distraught**

## Avoid these problems by identifying an owner

But who in a company is best suited to be this owner? Our research includes interviews with over 400 product managers, data teams, and engineers. Through this research, it became clear that **company size** was the biggest influencer on who took ownership of the tracking plan.



In **startups** we saw that most often a head of growth or head of product would take ownership – but process around tracking is very ad hoc. A company in such an early stage naturally will have less people involved in the tracking plan, therefore making it relatively easy to manage.

At **SME** size, the head of data/ analytics typically owns the tracking plan. A company at this stage of growth will have a wider audience of data and insight, so event tracking becomes more complex, and therefore there is a more definite need for ownership.

At the **enterprise** level, the ultimate owner is the head of product analytics. Companies at this scale will of course have the entire team managing and enforcing the tracking plan, with the team head bearing responsibility.

## We've seen these approaches work with varying degrees of success – which works best?

Our research has made clear that no matter the size or type of your organisation, the **product team** will be the most successful owner of your tracking plan. Who in the product team depends entirely on team size. Product managers could collaborate on this, or the sole owner could be a product analyst.

While some product managers may already naturally take ownership over the tracking plan, others may not buy-in to the idea right away. Empowering the right people to view tracking plan ownership as a key part of their role requires a bit of work.

### FOSTER A CULTURAL SHIFT

Raise the mentality of the company to understand adding tracking is a sub-task of building new features and products. Do not block the release of a new feature because it doesn't have tracking yet; instead, the implementation of tracking should be a part of the build process from the beginning: the PM should be creating Jira tickets for this just like anything else. Celebrate the performance and success of a new feature or product release, and not just the fact that it shipped.

### DEFINE CLEAR SUCCESS CRITERIA

This of course should happen before the tracking plan is implemented. Otherwise when tracking is underway, you won't know what good looks like. PMs should know what metrics will be best for measuring success, and analysts should be part of these conversations.

### COLLABORATE WITH SUPPORTING TEAMS

As owner of the tracking plan, a PM must make sure all relevant teams are involved. The engineering team will be the ones actually implementing it: do they have capacity? Is the proposed plan possible under the current infrastructure? Data/analyst teams also need to be heavily involved, because they will be the ones generating the reports. We've even seen some analysts practice structured 'office hours', so that other teams can benefit from their expertise on a regular basis.

### FINALLY, TRAIN TO SELF-SERVE

It's important to remember that technical training isn't always necessary. PMs don't need to know how to query a database – that's not their job. Nor should they (or anyone else) view the central team of analysts as an organisational resource for reports. Instead, PMs should be empowered to go and get insights when they need them. Tools like Amplitude and Mixpanel are great for this.

## Our recent case study with Beekeeper demonstrates the success of this approach to ownership



### THE CHALLENGE:

Beekeeper have multiple teams working across their iOS, Android, and web apps. All teams were responsible for implementing tracking on their respective features, and they relied on a large table in a Confluence page to keep an eye on this. The Confluence page quickly became outdated, and features went live without tracking.



### THE SOLUTION:

Switching sole ownership of the tracking plan to the product team resulted in new features going live with consistent, accurate, fully tested tracking. Plus the use of Iteratively has enabled the product team to trust their data again, and therefore leverage it more effectively.

## 5. Establishing processes

Defining key goals and identifying an owner are the first steps in creating a great tracking plan. The next step is to establish a repeatable process. In this section we will outline some best practices on how to do this. Ultimately, these processes will empower you to fully trust your data.

**"You can have the best charts, analysis, and data exploration tools but without the right data – useful, trustworthy, understandable, and accessible data – you'll struggle. A well managed taxonomy and lightweight governance are invisible force multipliers."**

– [Spenser Skates](#), CEO of Amplitude

### Deciding which events to capture

In the first section of this paper we discussed defining goals: if you know what you want to achieve, deciding on which events you want to track should be easy. You know your product better than anyone else. What do you want to learn from how people use it? What key things do you need to know before you can truly improve user experience and increase engagement?

The framing of these questions should help you figure out which events and properties should be tracked. Remember, try not to over-optimize too early – just start by tracking the events that interest you the most.

## Maintaining consistency

As mentioned before, building this tracking plan is a collaborative process. You'll have the developers who are actually going to implement it, and then the audiences for the data that comes out: analysts building reports, members of the product team who'd want to ask questions, and so on. So, once you've decided which events to capture, work together with these other teams to agree on some shared taxonomies.

So for instance, if your product is a music streaming service, you may want to track every time a user plays a song. What will this event actually look like in your tracking plan? Consider these three things:

- 1. Decide on a naming convention:** there are a number of ways you can label an event with 'song played'. It doesn't matter what naming convention you pick, as long as everyone sticks to it.

Some examples:

i.	camelCase	e.g. songPlayed
ii.	snake_case	e.g. song_played
iii.	Title Case	e.g. Song Played

- 2. Decide on which tense to use:** all events should be in the same tense for absolute clarity. We strongly recommend you use past tense ('Song Played', not 'Song Play'), because it's more accurate.
- 3. Use the object-action framework:** all events have an object, and the action that is performed on that object. So in this case, the object is 'song' and the action is 'played'. State the object first, then the action: 'Song Played'. As another example, if a user saves an album, it would be 'Album Saved'.

---

**Remember: alongside your events and their associated metrics should be properties which contextualise these events. The properties are also subject to the above guidelines. So, with 'Song Played', some properties might be 'songName' and 'userId'.**

---

## Determining where events should be captured

We've found that lots of companies limit themselves by only capturing events client-side. For the most reliable event tracking, and most useful analytics, we strongly recommend you capture events on both the server and client side. Let's take a quick look at why: So for instance, if your product is a music streaming service, you may want to track every time a user plays a song. What will this event actually look like in your tracking plan? Consider these three things:

**Client-side tracking** will provide much richer information; for instance, on the client-side you can see when users play or pause songs, and all the context surrounding that. But it's important to keep in mind that doing it all on the client-side presents you with limitations you can't control, like ad-blockers and browser restrictions.

**Server-side tracking** lacks richness and context, but is far more reliable. You can track things like if user authentication, or billing, is working properly on the server-side. Any mission-critical events should be tracked here.

## Implementing your plan

Now that you know **what** your plan is, you now need to actually implement it – the way you do this, as with everything else in this section, can be established as a repeatable process.

If you've created a tracking plan with the above processes in mind, it should be fairly straightforward to implement, because you know that all teams will be looking at the same thing. The key first steps in implementation are:

1. Prioritise what you're going to implement, based on the goals you identified at an earlier stage. There's no need to implement everything in one go, especially if all of these processes are completely new.
2. Make sure the events you've prioritised are then picked up by developers. This should be treated like any other sub-task of building a new feature or product; it gets a Jira ticket just like everything else. The important thing is building this into the regular workflow of the engineering team.

## Documenting your plan

It's very likely that when you sit down and put the tracking plan together, you may do it all in one go as a big group, or over a couple of days. Either way, what with establishing which events you're going to track and what your taxonomies are, it's likely that team members will forget the fine details. Documenting your tracking plan is really important; everyone can refer back to the documentation, and you can use it to on-board new team members.

**AS A MANUAL EXERCISE, THIS CAN BE FAIRLY TEDIOUS.**

**AS A BARE MINIMUM, DOCUMENT THE FOLLOWING:**

- |   |   |   |
|---|---|---|
| <ul style="list-style-type: none"><li>• Your framework and taxonomies: all the things you do to maintain consistency.</li></ul> | <ul style="list-style-type: none"><li>• The tracking plan itself: a complete list of all the events, properties, descriptions and other key information (can be using one of the templates we shared earlier or Iteratively).</li></ul> | <ul style="list-style-type: none"><li>• The instrumentation process: how a single event goes from an item on a list, to a task on a Jira ticket, and finally embedded into code, tested and tracked successfully.</li></ul> |
|---|---|---|

Whoever owns the tracking plan should stay on top of (or delegate) keeping this up to date. This is always going to be a challenge – keeping documentation relevant is one of those tasks that often gets neglected.

## Enforcing the plan

Having all of the above in place is great – but it will all be for nothing if none of it is consistently **enforced**. Unfortunately, just having accurate documentation is not a guaranteed way to make this happen.

### Iteratively enforces your tracking plan for you

A Google Sheet or Confluence page has its clear limitations; a tracking plan is much more than just a list of events and properties – it's a dynamic document, and should support and enforce the instrumentation of the plan itself.

With Iteratively you can replace that Google Sheet entirely and enforce your tracking plan in the product, ensuring you have trustworthy data in your downstream tools.

Our intuitive web app is where you work on the tracking plan: you can create new events and properties, update existing ones and ensure that

what's in the plan is actually implemented in the product. You can easily enforce a consistent event naming convention across all your products and data sources to avoid messy data downstream.

Our best-in-class developer tooling is used by your engineers to implement your plan. We generate tracking libraries on-the-fly, so developers can quickly and easily instrument new events using our command line app. And with strongly typed SDKs for every major development platform, you avoid typos, missing properties and wrong data types - think auto-complete for your analytics.

Learn more about Iteratively [here](#).



## 6. How to get started – a checklist

Hopefully by reading this white paper you understand that events tracking is an ongoing collaborative process, and as such is never 'done'. This set of guiding principles should help you establish that process:

### DATA IS A TEAM SPORT:

seeing as data touches everyone, everyone will have a hand in informing how it's used.

### DATA ITSELF ISN'T POWERFUL, THE WAY YOU USE IT IS:

you could have the most amazing tracking plan in place, feeding data elegantly into a reporting system – but none of it matters if you don't use it to make decisions.

### GOOD PROCESSES ARE SECOND NATURE:

if you're in the business of building software, you already know this. Austen Allred, the CEO of the Lambda School, made this observation when highlighting the extremely wide-spread adoption of the Git process:

**"One of the reasons software is so powerful is it forces operationally complex businesses to agree definitively on processes, and there's an established and agreed upon protocol for changing them."**

— [Austen Allred](#) on Twitter

**We built Iteratively with these principles in mind, which is why it's an excellent tool for building and enforcing a fully collaborative and accessible tracking plan.**

## What next? A checklist

If you're just getting started with this, we cannot recommend highly enough to only track the basics. Capturing, processing and storing "everything" will quickly become unsustainable and costly. Finding what you need in that much data will be like looking for a needle in a haystack. Take a practical approach and ask yourself: what's most important to our business?

### Try using this checklist:



**Identify key goals for the year**



**Prioritise these goals**



**From these goals identify just TEN QUESTIONS**



**Figure out what data is needed to answer those questions**



**Figure out what events and properties you need to track to get that data**



**Instrument those events and properties into your products**



**Ensure your tracking is instrumented according to your plan – can be done manually or using a tool like Iteratively**



**Document everything as you go and try to keep your plan up to date!**

Once you've got the process going with those ten questions, you'll already be in a great place. Iterate by continuing to add questions.

# Iteratively is here to help you

Overwhelmed by all the hazards and difficulties that come with designing, instrumenting and evolving your event tracking? We definitely were and that's exactly why we built Iteratively.

Iteratively helps data teams, product managers and engineers define, instrument, verify and collaborate on event tracking. We proactively solve the data quality issues that arise from inconsistent event naming and missing tracking and provide a workflow for managing the evolution of your tracking.

We help you ditch the spreadsheet, schematize your event data and enforce your tracking plan so you have high quality data to work with, no data munging required. If you're interested in trying out Iteratively for your company, [create an account today](#) or [book a demo with our team](#) to learn more.

