6

Organizational Metrics

> If you can't measure it, you can't improve it
> – *Peter Drucker (longer version by Lord Kelvin)*

> [Watermelon Metric:] . . .teams think they are doing a great job hitting green targets, their customers view it quite differently and only see red
> – *Barclay Rae (2014)*

> When optimizing for conversion, we often find clients trying to improve engine torque while ignoring a flat tire
> – *Bryan Eisenberg and John Quarto-vonTivadar (2008)*

Why you care? *Organizations that want to measure their progress and accountability need good metrics. For example, one popular way of running an organization is to use Objectives and Key Results (OKRs), where an Objective is a long-term goal, and the Key Results are shorter-term, measurable results that move towards the goal (Doerr 2018). When using the OKR system, good metrics are key to tracking progress towards those goals. Understanding the different types of organizational metrics, the important criteria that these metrics need to meet, how to create and evaluate these metrics, and the importance of iteration over time can help generate the insights needed to make data-informed decisions, regardless of whether you also run experiments.*

Metrics Taxonomy

In a data-driven organization, metrics and the accompanying data analyses can be used at every level, from top-level goal setting and accountability on down through the teams. The discussion of what the metrics should be for an

organization or a team is useful for aligning on goals, and subsequently providing transparency and accountability on executing towards those goals (Doerr 2018). This section focuses on organizational metrics overall, whereas Chapter 7 discusses metrics specific for experimentation and Chapter 21 discusses the role of guardrail metrics for alerting in experiments.

In discussing organizational metrics, the taxonomy commonly used is goals, drivers, and guardrails. This taxonomy is useful regardless of whether we are talking about an organization that is an entire company or a specific team within a larger organization.

Goal metrics, also called *success metrics* or *true north metrics*, reflect what the organization ultimately cares about. When trying to come up with a goal metric, we recommend first articulating what you want in words. Why does your product exist? What does success look like for your organization? The leaders of the organization must engage in answering these questions, and the answers are often tied to a mission statement. For example, if Microsoft's mission is to empower every person and every organization on the planet to achieve more, or Google's mission is to organize the world's information, then their goals are often directly related to those missions.

Being able to articulate your goal in words is important, as the transformation of that goal into metrics is often imperfect, and your goal metrics may be proxies of what you really care about and require iteration over time. Having people understand the limitations and differences between the metrics and the articulation of the goal is critical to driving the business in the right direction.

Goal metrics are usually a single or a very small set of metrics that best captures the ultimate success you are striving towards. These metrics may not be easy to move in the short term because each initiative may have only a very small impact on the metric, or because impacts take a long time to materialize.

Driver metrics, also called *sign post metrics*, *surrogate metrics*, *indirect* or *predictive metrics*, tend to be shorter-term, faster-moving, and more-sensitive metrics than goal metrics. Driver metrics reflect a mental causal model of what it takes for the organization to succeed, that is, hypotheses on the drivers of success rather than just what success looks like.

There are several useful metrics frameworks for thinking about what drives success: The HEART framework (Happiness, Engagement, Adoption, Retention, and Task Success) (Rodden, Hutchinson and Fu 2010), Dave McClure's PIRATE framework (*AARRR! Acquisition, Activation, Retention, Referral, Revenue*) (McClure 2007), or user funnels in general. These frameworks can help break down the steps that lead to success. For example, before ultimately achieving revenue, a typical company must acquire users and ensure that their product is engaging enough to retain them.

A good driver metric indicates that we are moving in the right direction to move the goal metric(s).

Guardrail metrics guard against violated assumptions and come in two types: metrics that protect the business and metrics that assess the trustworthiness and internal validity of experiment results. Here, we focus on the first type of organizational guardrails, while trustworthiness guardrail metrics are discussed in Chapter 21.

While our eyes are usually on the goal and driver metrics, guardrail metrics are important to ensure we move towards success with the right balance and without violating important constraints. For example, our goal may be to get as many users as possible to register, but we don't want the per-user engagement level to drop drastically. Another example is a password management company. There might be a tradeoff between security (no hijackings or information stolen), ease-of-use, and accessibility (i.e, how often users are locked out). While security may be the goal, the ease-of-use and accessibility can be guardrails. Finally, while page-load-time may not be a goal metric, we still need to make sure that feature launches do not degrade load times (see Chapter 5). Guardrail metrics are frequently more sensitive than goal or driver metrics. See Chapter 21 for more examples of guardrail metrics.

While we find goal, driver, and guardrail metrics offer the right amount of granularity and comprehensiveness, there are other business metric taxonomies as well:

- **Asset vs. engagement metrics**: Asset metrics measure the accumulation of static assets, like the total number of Facebook users (accounts) or total number of connections. Engagement metrics measure the value a user receives as a result of an action or by others using the product, such as a session or a pageview.
- **Business vs. operational metrics:** Business metrics, such as revenue- per-user or daily active user (DAU), track the health of the business. Operational metrics, such as queries per second, track whether there are operational concerns.

While we discuss metrics for experiments further in Chapter 7, there are also other types of metrics commonly used in experimentation. **Data quality metrics** ensure the internal validity and trustworthiness of the underlying experiments (see also Chapter 3 and Chapter 21). **Diagnosis or debug metrics** are helpful when debugging a scenario where the goal, driver, or guardrail metrics indicate there is a problem. They might provide additional granularity or other information typically too detailed to track on an ongoing basis but

useful when drilling down into a situation. For example, if click-through rate (CTR) is a key metric, you might have 20 metrics to indicate clicks on certain areas of the page. Or, if revenue is a key metric, you might want to decompose revenue into two metrics: a revenue indicator that is a Boolean (0/1) indicating whether the user purchased at all; and a Conditional Revenue metric that comprises the revenue if the user purchased and is null otherwise (when averaged, only the revenue from purchasing users is averaged). Average overall revenue is the product of these two metrics, but each tells a different story about revenue. Did it increase/decrease because more/less people purchased or because the average purchase price changed?

Regardless of the taxonomy used, having discussions on metrics is useful, as agreeing on metrics requires clear goal articulation and alignment. The metrics can subsequently be used for goal setting at the company level, team level, feature level or individual level, and be used for everything from executive reporting to engineering system monitoring. Iterating on metrics over time is also expected, both as the organization evolves and the understanding of the metrics evolves.

We often need to measure goals, drivers, and guardrails at both the company level and team level. Each team is likely to contribute differently to the overall success of the company. Some teams might be more focused on adoption, others on happiness, still others on retention or performance or latency. Each team must articulate their goal and hypothesis on how their metrics relate to the overall company metrics. The same metric may play a different role for different teams. Some teams may use latency or other performance metrics as a guardrail, while an infrastructure team may use those same latency or performance metrics as their goal metric and use the other business metrics as their guardrail metrics.

For example, let's say you are working on a product where the overall goal metric is long-term revenue, and driver metrics at a business level are user engagement and retention. Now, you have a team that is working on a support site for this product. This team tried to set "time-on-site" as the key driver metric to improve, but is more time on the site better or worse? This type of discussion is useful at every level of the company to understand and align on.

Parmenter in *Key Performance Indicators* (2015) uses the diagram shown in Figure 6.1 to emphasize the importance of aligning goal and driver metrics to your overall business strategy.

Depending on organization size and objectives, you may have multiple teams, each with their own goal, driver, and guardrail metrics, and all of which must align with your overall goal, driver, guardrail metrics.

Figure 6.1 It is important to align each team's metrics with the overall goal and strategic direction

Formulating Metrics: Principles and Techniques

Now that you have it down in words what success looks like and possible drivers, let's start formulating metrics. This is when we take a qualitative concept to a concrete, quantifiable definition. In some cases, such as revenue, the answer may be obvious. However, a company may define success as long-term revenue, which is harder to measure than revenue realized today. Other difficult-to-measure concepts of success include user happiness and user trust.

Key principles when developing goal and driver metrics are:

1. Ensure that your goal metrics are:
   - **Simple**: easily understood and broadly accepted by stakeholders.
   - **Stable:** it should not be necessary to update goal metrics every time you launch a new feature.
2. Ensure that driver metrics are:
   - **Aligned with the goal**: It is important to validate that the driver metrics are in fact drivers of success. One common technique for this validation is to run experiments expressly for this purpose. We discuss this further below.
   - **Actionable and relevant**: Teams must feel that they can act on the levers (e.g., product features) to move these metrics.
   - **Sensitive**: Driver metrics are leading indicators for goal metrics. Ensure that they are sensitive enough to measure impact from most initiatives.
   - **Resistant to gaming**: Because driver metrics and your goal metrics measure success, don't make them easily gameable. Think through the incentives and what behavior a metric may drive and how it might be gamed. See *Sidebar: Gameability* later in this chapter.

With these principles in mind, here are some helpful techniques and considerations for developing metrics:

- Use hypotheses from less-scalable methods to generate ideas, and then validate them in scalable data analyses to determine a precise definition

(see Chapter 10). For example, user happiness or user task success might only be directly measurable through user surveys, a methodology that is not scalable. However, we can conduct surveys or user experience research (UER) studies (see Chapter 10) to observe the types of behavior typically correlated with success and happiness. You can explore those behavior patterns using online logs data analysis at scale to determine whether those metrics work as a high-level metric. One concrete example is bounce rate, which is the proportion of users that stay only a short time on a website. We may notice that a short stay correlates with dissatisfaction. Combining that observation with a data analysis helps determine the exact threshold (should the threshold be 1 pageview? 20 seconds?) needed to precisely define the metric (Dmitriev and Wu 2016, Huang, White and Dumais 2012).

- Consider *quality* when defining goal or driver metrics. A click on a search result is a "bad" click if the user clicks the back button right away; a new user signup is a "good" signup if the user actively engages with the website; a LinkedIn profile is a "good" profile if it contains sufficient information to represent the user, such as education history or current and past positions. Building a quality concept, such as with human evaluation (see Chapter 10), into your goal and driver metrics makes it much more likely that movement from these metrics leads to a solid interpretation on which to base decisions.
- When incorporating statistical models in the definition of a metric, it is essential to keep the model interpretable and validated over time. For instance, to measure long-term revenue from a subscription, it is common to compute the lifetime value (LTV) based on predicted survival probability. However, if the survival function is too complicated, it may be hard to get buy-in from stakeholders, even harder if a sudden drop on the metric needs to be investigated. Another example is Netflix using bucketized watch hours as driver metrics because they are interpretable and indicative of long-term user retention (Xie and Aurisset 2016).
- Sometimes it may be easier to precisely measure what you do not want, such as user dissatisfaction or unhappiness, than it is to measure what you want. For example, how long does a user have to stay on a site to be considered "satisfied?" On sites with tasks, like search engines, a short visit to a site pointed to from a search result is more often correlated with a user being unhappy than a long visit. That said, a long visit can imply either that a user is finding what they need or that they are trying hard to do something and in fact getting frustrated. In this way, *negative* metrics are useful as guardrail or debug metrics.
- Always remember that metrics are themselves proxies; each has its own set of failure cases. For example, a search engine may want to use CTR to

measure user engagement but driving just CTR may lead to increased clickbait. In such cases, you must create additional metrics to measure the edge cases. In this example, one possibility is to use human evaluation (see Chapter 10) as a metric to measure relevance and counterbalance a tendency towards rewarding clickbait.

Evaluating Metrics

We have outlined several principles to follow when developing metrics. Most metrics evaluation and validation happen during the formulation phase, but there is work that needs to be done over time and continuously. For example, before adding a new metric, evaluate whether it provides additional information compared to your existing metrics. Lifetime value (LTV) metrics must be evaluated over time to ensure that prediction errors stay small. Metrics heavily relied on for experimentation must be evaluated periodically to determine whether they encouraged gaming (i.e., whether a threshold used in a metric definition cause disproportional focus on moving users across the threshold).

One of the most common and challenging evaluations is establishing the causal relationship of driver metrics to organizational goal metrics, that is, whether this driver metric really drives the goal metrics. In for-profit organizations, Kaplan and Norton wrote "Ultimately, causal paths from all the measures on a scorecard should be linked to financial objectives" (Kaplan and Norton 1996). Hauser and Katz (Hauser and Katz 1998) write, "the firm must identify metrics that the team can affect today, but which, ultimately, will affect the firm's long-term goals." Spitzer (Spitzer 2007) wrote that "measurement frameworks are initially composed of hypotheses (assumptions) of the key measures and their causal relationships. These hypotheses are then tested with actual data, and can be confirmed, disconfirmed, or modified." This characteristic is the hardest to satisfy, as we often don't know the underlying causal model, and merely have a hypothesized mental causal model.

Here are a few high-level approaches to tackle causal validation that you can also apply to other types of metrics evaluation:

- Utilize other data sources such as surveys, focus groups, or user experience research (UER) studies to check whether they all point in the same direction.
- Analyze observational data. While it is difficult to establish causal relationships with observational data (as we discuss in Chapter 11), a carefully conducted observational study can help invalidate hypotheses.

- Check whether similar validation is done at other companies. For instance, several companies have shared studies that show how site speed impacts revenue and user engagement (see Chapter 5). Another example is studies that show the impact of app size on app downloads (Reinhardt 2016, Tolomei 2017).
- Conduct an experiment with a primary goal of evaluating metrics. For example, to determine whether a customer loyalty program increases customer retention and therefore customer LTV, run experiments that slowly rollout the customer loyalty program, and measure retention and customer LTV. We caution that these experiments often test a relatively narrow hypothesis, so it still requires work to generalize the results.
- Use a corpus of historical experiments as "golden" samples for evaluating new metrics. It is important that these experiments are well understood and trustworthy. We can use these historical experiments to check for sensitivity and causal alignment (Dmitriev and Wu 2016).

Note that the challenge of relating driver metrics to goal metrics also applies for guardrail metrics. See our example in Chapter 5 of how to conduct an experiment to measure the impact of latency, a guardrail metric, on goal metrics.

Evolving Metrics

Metric definitions evolve over time. Even if the concept stays the same, the exact definition may still change. Change can happen because:

- The business evolved: The business may have grown and created new business lines. This could lead to the business changing its focus, such as shifting from adoption to engagement and retention. One specific type of evolution to call out is a shift in user base. When calculating metrics or running experiments, note that all of that data is coming from the existing user base. Especially for early-stage products or start-ups, early adopters may not be representative of the user base that a business desires in the long-term (Forte 2019).
- The environment evolved: The competitive landscape may have changed, more users may be aware of privacy concerns, or new government policies may be in effect. All of these changes can shift the business focus or perspective, and therefore what you measure with metrics.
- Your understanding of the metrics evolved: Even metrics you carefully evaluated during the development phase, when observing its performance

in action (e.g., looking for gameability), you may discover areas of improvement that leads to more granularity or different metric formulations. Hubbard (Hubbard 2014) discusses Expected Value of Information (EVI), which is a concept that captures how additional information helps you make decisions. Taking the time and effort to investigate metrics and modify existing metrics has high EVI. It is not enough to be agile and to measure, you must make sure your metrics guide you in the right direction.

Certain metrics may evolve more quickly than others. For example, driver, guardrail, and data quality metrics may evolve more quickly than goal metrics, often because those are driven by methodology improvements rather than fundamental business or environmental evolutions.

Because metrics will evolve over time, you should become more structured in handling changes in metrics as your organization grows. Specifically, you will need infrastructure to support the evaluation of new metrics, the associated schema changes, backfilling of data needed, and more.

Additional Resources

There are several great books about metrics, measurements, and performance indicators (Spitzer 2007, Parmenter 2015, McChesney, Covey and Huling 2012). Spitzer notes that "What makes measurement so potent is its capacity to instigate informed action—to provide the opportunity for people to engage in the right behavior at the right time." In the context of controlled experiments, because the Treatment is the *cause* of the impact to each metric (with high probability for highly statistically significant effects), formulating the key metrics is an assessment of the value of an idea (the Treatment) on some axis of interest.

SIDEBAR: Guardrail Metrics

There are two types of guardrail metrics: trustworthiness-related guardrail metrics and organizational guardrail metrics. Trustworthiness-related guardrail metrics are discussed in detail in Chapter 21, as those are necessary to ensure that experimental results are trustworthy. Here we discuss organizational guardrail metrics.

As we discuss in Chapter 5, an increase in latency of even a few milliseconds can result in revenue loss and a reduction in user satisfaction. Thus,

latency is often used as a guardrail metric because it is so sensitive, especially relative to revenue and user satisfaction metrics. Most teams are typically working on new features that are trying to move goal or driver metrics but, in doing so, they check latency and try to ensure that their feature does not increase latency. If it does, then that triggers a discussion about tradeoffs such as whether the impact of the new feature is worth the impact from the increase in latency, whether there are ways to mitigate the increase, or whether there are ways to offset the new feature with other features that improve (decrease) latency.

Many organizational guardrail metrics are similar to latency, sensitive metrics that measure phenomena known to impact the goal or driver metrics, but that most teams should not be affecting. Examples of such metrics include:

1. HTML response size per page. On a website, the server response size is an early indicator that a large amount of code (such as JavaScript) was introduced. Alerting on such a change is a great way to uncover a possibly sloppy piece of code that could be optimized.
2. JavaScript errors per page. Degrading (i.e., increasing) the average number of errors on the page is a ship blocker. Segmenting by browsers helps to identify whether the JavaScript issue is browser dependent.
3. Revenue-per-user. A team that works on one part of the product, such as relevance, may not realize that they are hurting revenue. Revenue-per-user usually has high statistical variance, so it is not sensitive as a guardrail; more sensitive variants can be great alternatives, such as revenue indicator-per-user (was there revenue for user: yes/no), capped revenue-per-user (anything over $X is capped to $X), and revenue-per-page (there are more page units, although care must be taken to correctly compute the variance, see Chapter 22).
4. Pageviews-per-user. Because many metrics are measured per page (such as, CTR), a change to pageviews-per-user could imply that many metrics changed. It is natural to focus on the numerator, but if pageviews-per-user changes, it is the denominator that changes, which requires thought. If the change is unexpected, it is worth reviewing the reasons carefully (Dmitriev et al. 2017). Note that pageviews-per-user may not work as a guardrail in all cases; for example, if you are testing an infinite scroll feature, then pageviews-per-user will almost certainly change.
5. Client crashes. For client software (e.g., Office Word/PowerPoint/Excel, Adobe Reader) or phone applications (e.g., Facebook, LinkedIn, Minecraft, Netflix), crash rate is a critical guardrail metric. In addition to a count metric (crashes-per-user), an indicator is commonly used (Did the user crash

during the experiment?), which is averaged over all users, as indicators have lower variance and thus show statistical significance earlier.

Different teams may swap which metrics are their goal, driver, and guardrail metrics. For example, while most teams may use the canonical goal, driver, and guardrail metrics, an infrastructure team, for example, may use performance or organizational guardrail metrics as their goal (and use the product team's goal and driver metrics as their guardrails). Just like driver metrics, it is important to establish the causal relationship between guardrail metrics and goal metrics, as was done in Chapter 5.

SIDEBAR: Gameability

Your goal and driver metrics need to be hard to game: when given a numerical target, humans can be quite ingenious, especially when the measures are tied to rewards. There are numerous examples throughout history:

- Vasili Alexeyev, a famous Russian super-heavyweight weightlifter, was offered an incentive for every world record he broke. The result of this contingent measurement was that he kept breaking world records a gram or two at a time to maximize his reward payout (Spitzer 2007).
- A manager of a fast-food restaurant strived to achieve an award for attaining a perfect 100 percent on the restaurant's "chicken efficiency" measure (the ratio of how many pieces of chicken sold to the number thrown away). He did so by waiting until the chicken was ordered before cooking it. He won the award but drove the restaurant out of business because of the long wait times (Spitzer 2007).
- A company paid bonuses to its central warehouse spare parts personnel for maintaining low inventory. As a result, necessary spare parts were not available in the warehouse, and operations had to be shut down until the parts could be ordered and delivered (Spitzer 2007).
- Managers at a hospital in the United Kingdom were concerned about the time it was taking to treat patients in the accident and emergency department. They decided to measure the time from patient registration to being seen by a house doctor. The nursing staff thus began asking the paramedics to leave their patients in the ambulance until a house doctor was ready to see them, thus improving the "average time it took to treat patients" (Parmenter 2015).
- In Hanoi, under French colonial rule, a program paying people a bounty for each rat tail handed in was intended to exterminate rats. Instead, it led to the

farming of rats (Vann 2003). A similar example, although likely anecdotal, is mentioned with regards to cobra snakes, where presumably the British government offered bounty for every dead cobra in Delhi and enterprising people began to breed cobras for the income (Wikipedia contributors, Cobra Effect 2019).

- Between 1945 and 1960, the federal Canadian government paid 70 cents a day per orphan to orphanages, and psychiatric hospitals received $2.25 per day, per patient. Allegedly, up to 20,000 orphaned children were falsely certified as mentally ill so the Catholic Church could get $2.25 per day, per patient (Wikipedia contributors, Data dredging 2019).
- Funding fire departments by the number of fire calls made is intended to reward the fire departments that do the most work. However, it may discourage them from fire-prevention activities that reduce the number of fires (Wikipedia contributors, Perverse Incentive 2019).

While these examples show the importance of choosing metrics carefully, how does this apply in the online domain? One common scenario is to use short-term revenue as a key metric. However, you could increase short-term revenues by raising prices or plastering a website with ads, and either of those would likely lead to users abandoning the site and customer LTV declining. Customer LTV is a useful guiding principle when considering metrics. More generally, many unconstrained metrics are gameable. A metric that measures ad revenue *constrained* to space on the page or to a measure of quality is a much better metric to ensure a high-quality user experience. How many queries return no results is gameable without some quality constraint because one can always return bad results.

Generally, we recommend using metrics that measure user value and actions. You should avoid vanity metrics that indicate a count of your actions, which users often ignore (the count of banner ads is a vanity metric, whereas clicks on ads indicates potential user interest). At Facebook, creating user "Likes" is an example where there is a UI feature that both captures user actions and is a fundamental part of the user experience.