

OKRs are Bullshit



DRMORR
FEB 2, 2024



Share



It's a new year, time for a new rant [1](#)! And yes, before you ask, the post title is deliberately provocative. You might say this is my ploy to get more paid subscribers, because only paid subscribers can leave comments and I expect that the title alone will make many of you want to comment. 😏

Want to read more unhinged rants? Want to leave comments about how unhinged I am? Subscribe now!

Type your email...	Subscribe
--------------------	-----------

Anyways, I expect that many of my readers just finished up their quarterly (and/or yearly) planning cycle, so I thought this would be a good time to remind you all that the process we've all settled on in the tech industry is nonsense: I am, of course, referring to the [Objectives and Key Results](#) framework. So let's talk about OKRs, what they are and where they come from, and why they're a terrible idea [2](#).

OKRs: A Google Conspiracy?

The OKR framework was originally developed by Google back in—

Wait a minute, I just read the Wikipedia article I linked in the previous section, and it turns out I'm starting this off not only by being rude, but also spreading misinformation! How could I. Let's try this again.

OKRs were introduced by Andrew Grove at Intel, all the way back in the 1970s! He wrote about them in a book on management in 1983, and later they were introduced at Google, I guess sometime in the early 2000s. And while Google didn't *invent* the concept of OKRs, Google certainly helped popularize them [3](#). Now it doesn't matter where you go, every company has OKRs. The term has become like "Kleenex"—it's used ubiquitously to mean "planning", regardless of how similar or not the planning process actually is to the original OKR framework [4](#).

So with the backstory out of the way, what *are* OKRs? In short, they're a way of goal-setting and then measuring your progress towards the goals. The "Objective" is your goal, and the "Key Results" are the things you need to accomplish to know whether you've hit your goal. Of course because we want to be data-driven organizations, the key results need to be measurable and metrics-based.

Typically, OKRs are supposed to be cascading. In other words, the CEO (or whoever's in charge) sets some OKRs for the organization as a whole, and then the individual business units set OKRs that support the global OKRs, and then each team sets OKRs that support the business unit's, and (potentially) each team member sets their own personal OKRs. At each level, you should have between one and three objectives, which are short statements about "what" you want to accomplish in the next quarter, or year, or whatever, and each objective should have between one and three key results which indicate the success or failure of the objective.

In addition to the core framework, there are a few guiding principles that organizations should use when setting OKRs. Most (in)famously, you should set your OKRs so you only achieve 70% of them. If you're consistently hitting 100% on your goals, that means you're not being ambitious enough. Secondly, you should avoid "binary" OKRs, that is, OKRs whose only metric is "I did the thing" or "I didn't do the thing". Thirdly, OKRs aren't supposed to encompass all of your organization's activities: normal, day-to-day maintenance work, on-call support, etc. are "extra" things that don't get captured by your OKRs [5](#). And lastly, the only way to learn OKRs is by doing OKRs.

Now, some of you are all prepared to whip out your credit cards and subscribe so you that you can angrily tell me that I've got it all wrong and that I don't understand the framework at all. That's fine—I'm happy to have you as a subscriber, but I think gets at my fundamental complaint about the OKR framework: if the "only way to learn OKRs is by doing OKRs", then by definition everybody is gonna do OKRs differently, which means that in practice the framework becomes whatever you want it to become. But then,

when anybody comes out with *any* criticism of the OKR process, the response is always, in classic "[no true Scotsman](#)" style, "well, you're just not doing OKRs correctly." But I guess my question is: if nobody in the industry does OKRs "correctly", why are we still trying to do them at all?

Now look: I'm not arguing that we shouldn't have goals. I'm not arguing that we shouldn't make plans and try to hold ourselves accountable to those plans. We absolutely should! Engineers like to rage against process, bureaucracy, and friction, but I'll be the first to tell you that—especially in larger organizations—*some* process is important. My only point in this article is to hopefully convince you that OKRs ain't it.

OKRs: the road to a lot of unfinished work

So let's talk about the problems with OKRs. I want to preface this section by saying that my background is an infra engineer, and a lot of the points I make come from that perspective. But I've heard enough similar complaints from product people that I think my objections are valid in that setting as well.

First of all, let's start with the frankly ridiculous claim that you should target 70% completion for your OKRs. Setting aside the fact that this is very nebulous (should you complete 70% of your goals to 100%? Or should you complete 100% of your goals at 70%?) consider that much of the work we do doesn't actually have any value unless you do it *all the way*. Now maybe if your key result was "increase clickthrough rate by 100%" and you only increased it by 70%, you could argue that is still pretty good. But if your key result is "migrate 100% of users to the new system" and you only migrate 70%, guess what? Now you're stuck maintaining two systems in perpetuity. Fortunately, I haven't heard people espouse this tenet as much lately—I think people are realizing that it incentivizes the wrong things.

But this leads us straight into the second problem with OKRs: actually measuring things. Some people might argue that the migration example I used above is actually bad because it's a binary OKR—either you migrated or

you didn't. This leads to all kinds of contortions to develop a metric that still says "I migrated the thing" but isn't binary. Maybe you interview your customers and you want 100% of them to be happy on the new thing, but you'll count it as a success if only 70% of them are happy. Or maybe you measure the number of outages caused by the new thing, and your goal is "zero outages" [6](#).

However, there are additional problems here: one is that you just invented a bunch of extra work for yourself, because chances are whatever metric you concocted to measure your migration success didn't exist before: so you have to go build some tooling to collect the metric before you can even start working on the actual thing you care about—tooling and metrics that will probably languish and be forgotten about in a quarter or two after priorities change. Another is that often, the metrics you invent have no relation to the work you're doing—the happiness (or not) of your users probably has between five and zero percent to do with how good a job you did on the migration, and is 90% related to whether or not the new system was well-designed by somebody else who probably isn't even at the company anymore. A third is that some of these metrics are really hard to reason about. For example, in the "number of outages" metric, your target value is 0, which means that if you have *any outages at all* your score for that key result is undefined. You have to divide the number of outages you had by zero to get your percentage. Congratulations! Your metric value is whatever you want it to be!

I think the biggest problem with OKR's laser focus on measurement, though, is that *not everything should be measured*, even if you can! Being "data-driven" is a huge buzzword in the industry. We want to improve, we want to see how much we improved by, and then we want to tell the world how much we improved by so our stock price goes up. But there's a tremendous amount of work that shouldn't or can't be measured, or is very easy to misinterpret even if you *can* measure it. I think this article by Richard Marmorstein sums it up really nicely: [be good-argument driven, not data-driven](#). Being data-driven requires a) that you have the metrics, b) that you

know enough statistics to interpret the metrics correctly, and c) that you don't care about anything that can't be measured.

The last complaint I have about OKRs comes from their cascading nature. As an industry, we mostly rejected waterfall-style development a long time ago, and then promptly introduced a planning framework that encourages waterfall-style development. There's no room in the OKR framework for research or experimentation (because how do you measure research?), so you have to know what you want to do in excruciating detail at the point when you write down your OKR, because otherwise something might come up that prevents you from completing (or even getting 70%) on your OKR. But raise your hand if you've ever written down all your OKRs and then two months into the cycle, something comes up that obsoletes all of your goals.

"But wait, you're just doing it wrong!" I can hear you exclaim from here. "You're supposed to be agile! OKRs can change! You should react to new information!" Right, yep, I've heard that one before. But I can guarantee you that come performance review time, the people who decide whether you're being successful or not as an engineer are going to grade you on your original goals for the year, and if you have to change them it's going to be viewed as a failure. I mean, maybe this doesn't happen *everywhere*, but it will require a significant amount of cultural backpressure to prevent this outcome. So maybe just let's use a planning process that actually has room for change built in, instead of trying to shoehorn in one that just doesn't work.

OKRs: that means we need another spreadsheet, right?

You know what I didn't talk about at all in this blog post? Spreadsheets. Nowhere in the OKR framework does it say that you should list all your objectives and key results in a spreadsheet, and then check in on the metrics every month by updating some values in the spreadsheet. Nobody ever said that you should have a JIRA epic for your objectives, and then track all your

tickets by which OKR they belong to. Nobody ever said anything about "internal OKRs" versus "external OKRs" or roadmaps or planning meetings or... the list goes on.

And yet, my prediction is that every single manager in existence, as soon as they hear "OKR" will immediately think "spreadsheet" ⁷. And I think that's a problem too. See, as an industry, we've conflated "OKRs" with "planning", when I don't think they should be conflated at all. Even if you brush aside all the problems I pointed out with OKRs in the previous section, and go back to the original (or at least, "original" as "made popular by Google") definition, the purpose of OKRs is to be aspirational. That's where the whole 70% thing comes from in the first place. We want to set hard goals that will inspire people to do their best work, and then recognize that the goals were hard and not penalize people for failing to meet them 100% of the way.

And honestly? When taken through that lens, I love OKRs ⁸! We *should* be trying to do hard things, and we shouldn't be punishing folks when they fail at them. And, also: we *should* have a plan, and we should understand the work that we're going to be doing over the next few weeks-to-months, and *maybe* we need a spreadsheet or something to help manage that plan. But please, for the love of god, let's stop trying to shove metrics into our goal-setting framework, let's stop shoving our goal-setting framework into our quarterly planning process, and let's stop spending months on end planning only to have the whole thing upended two days into the cycle.

Anyways, that's all I've got for now. I promise next week I'll be less inflammatory.

Thanks for reading,

~drmorrr

¹ Some of my readers may recognize this blog post as a redux of a post I made internally at a previous employer. Yes, I used the same title. I was told after the fact that my post was the impetus for upper management making some changes

to the planning process—I don't have any way of independently verifying that statement, but I'm choosing to believe it because it means that sometimes the things I say have an impact.

- 2 I am continuing my goal of converting some of you to paid subscribers by making the section headlines deliberately provocative as well.
- 3 By the way, the title of this section refers to an idea I've seen espoused a few times that Google deliberately promoted the OKR framework as a way to sabotage the rest of the industry and make them less effective. I think there are a number of maxims we can apply to this idea, including [Betteridge's law](#), [Hanlon's Razor](#), [Occam's Razor](#), and I'm sure some others as well.
- 4 I'll leave the remainder of the Kleenex comparison to your imagination.
- 5 Daily grunt-work isn't "inspirational" enough and you're supposed to use OKRs to inspire people.
- 6 I've seen both of these approaches taken at various times, along with a bunch of other techniques. You'd be amazed at how creative people can get trying to map a binary variable onto a continuum.
- 7 Dear managers, I ♥ you. No, seriously, you do a really freaking hard job, and it's not a job I ever want. So thank you. Even if you do use a few too many spreadsheets.
- 8 Now there's a statement I never expected I would utter.



2 Likes

Comments



Write a comment...

© 2024 David Morrison • [Privacy](#) • [Terms](#) • [Collection notice](#)
[Substack](#) is the home for great writing