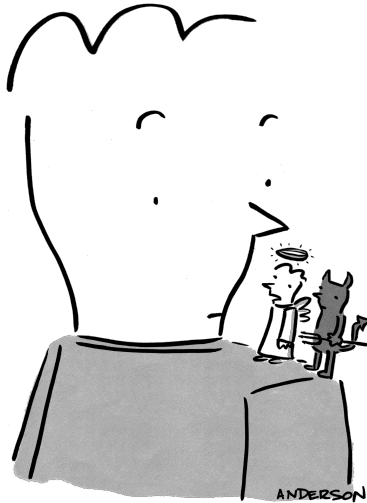


# Binstack: Making a maximal multi-dimensional decision

by Jason Cohen on July 2, 2022

Binstack is a technique for selecting the “single most impactful” solution when there are multiple, incomparable dimensions to evaluate.



“Do it, don’t do it. We don’t care anymore.”

Many decisions in life and business are instances of “multi-dimensional maximization,” in which we wish to pick the “single best” among a set of choices, but we’re confounded because each choice is variously better or worse along different dimensions.

Examples:

- Which major feature should we spend the next six months building?  
*(P would generate revenue, but Q would reduce cancellations, but R would save us money)*
- Which candidate should we hire?  
*(P has the best skills, but Q has more experience in our market, but R seems like the best culture-fit)*
- Which new marketing campaign should we spend thousands of dollars to test?  
*(P is cheaper to try, but Q has a larger reach, but R is targeted at our industry)*

Not only do you need the best decision, you also need to be able to **explain your decision to others**, especially to those who wish the decision had gone a different way. Do not under-value the importance of crisp explanation.

The “rubric” is the typical framework for these decisions; a separate paper on this site explains how to use one effectively for “ROI-style” decisions<sup>1</sup>. Unfortunately, while it may feel productive to fill many cells with many numbers, and while it may feel analytically rigorous to convolute those numbers into a final score, this fails to clearly identify the best choice, and fails to create a clear explanation for the choice.

<sup>1</sup> The goal of ROI is to maximize efficiency, i.e. deliver the most amount of value per unit of time. This paper asks a different question: How to decide the single most valuable thing, regardless of cost, with incommensurate and conflicting dimensions

After demonstrating and analyzing the causes of this failure, I present an alternative framework I’ve nick-named “Binstack<sup>2</sup>.”

<sup>2</sup> Additional resource: [Adam Waselnuk](#) created a [Notion Process Template for Binstack](#).

## Why rubrics don’t add up

Consider two players in a game, with attributes:

Attribute	Player P	Player Q
Health	8	5
Strength	5	7
Speed	2	6
Endurance	5	2

Which player is better? If one scored higher than the other in every dimension, the choice would be simple. In this case, each player is better than the other along two dimensions, and worse along two; objectively there’s no clear winner.

So let's try a rubric. In its simplest form, we add up the scores contributed by each dimension, and the total score decides the winner. Unfortunately, this doesn't tell us which one is better:

Attribute	Player P	Player Q
Health	8	5
Strength	5	7
Speed	2	6
Endurance	5	2
<b>Total:</b>	<b>20</b>	<b>20</b>

Games often engineer this result on purpose, to create players that are different but not over-powered. This creates a balanced game, but to make a confident decision about which one is "best," we need something imbalanced.

Real-life rubrics often result in a pile of options that share a similar score, resulting in no clear winner. Even if we make a decision, we can't explain the decision to others, because in actual fact it's a tie, and the tie was broken arbitrarily. That's no way to make a decision.

To create separation, people often add "weights" to the raw value to create a new kind of "score." In this case, suppose that with our game-playing style we don't care much about Speed, but we can really leverage Strength. So we assign weights which we multiply against the scores, to compute a customized metric of "value."

Attribute	Orig P	Orig Q	Wt	Wted P	Wted Q
Health	8	5	1.0	8	5
Strength	5	7	2.0	10	14
Speed	2	6	0.5	1	3
Endurance	5	2	1.0	5	2
<b>Total:</b>	<b>20</b>	<b>20</b>		<b>24</b>	<b>24</b>

Well, shoot. That didn't help. What if we contrive to *force* Q to be better, by intentionally using weights that penalize the two attributes where P is superior?

Attribute	Orig P	Orig Q	Wt	Wted P	Wted Q
Health	8	5	0.5	4	2.5
Strength	5	7	2.0	10	14
Speed	2	6	0.5	1	3
Endurance	5	2	0.5	2.5	1
<b>Total:</b>	<b>20</b>	<b>20</b>		<b>17.5</b>	<b>20.5</b>

Even with a conspiracy to throw the election for Q, it wins by a mere 2.5 points out of 20—hardly a resounding victory that would give everyone confidence in the decision.

This happens in the real world. Even with weights, a clear winner often does not emerge, and again we're back to a weak, indefensible decision.

Worse: In the real world we rarely have precise scores. Attributes like "potential new revenue" and "increased customer satisfaction" are not predictable with accuracy. For more qualitative measures we use scales "from 1 to 5" which are even less precise. This imprecision creates inherent error, which is then compounded by multiplying weights. Differences in the final results might be more error than signal. How would you even know?

Worse again: The attributes aren't comparable. Whatever units the game uses for "Health," they're not related to whatever units the game uses for "Speed." By adding them together, we're implicitly saying "these are comparable," but they are not. Weights are supposed to solve this by converting everything into some sort of "value," but if you say out loud what the weights are doing, it sounds incorrect. For example: "Every \$150k of additional ARR is exactly as valuable to us as 10% more customer satisfaction. Indeed, we would be fine with customer satisfaction going down 10% if we added \$300k in new ARR." Really?

It's largely noise, which is why we're unhappy with the so-called "winner."

## Solution: Binstack: Stack-ranked binary attributes

To transcend the noise, stop pretending that the values in the rows are precise or comparable.

A general rule of complex decisions, is that often they're difficult because we've avoided making other, smaller decisions. That is the case with the rubric, as we're about to fix.

Our top-level purpose is to pick the item that maximizes impact, and to be able to explain why. So, first we're going to make smaller decisions about the true impact of each item, and then we're going to make a decision about which impacts are most valuable.

### Binary materiality

No more values, no more weights, no more scores. Either an item *materially* contributes to that attribute, or it doesn't. "Materially" means the effect is so large you can measure it easily:

- Not just "more revenue," but at least a 10% bump so that the curve visibly changes.
- Not just "more retention," but a 20% decrease in cancellations tied to a specific cause.
- Not just "more intuitive," but a 40% decrease in support tickets on a certain topic.
- Not just "more competitive," but sales will add it to their standard presentation and marketing will add it to the feature-table on the pricing page.
- Not just "more profitable," but overall gross profit margin will improve by 1%.
- Not just "will pay for it," but putting it in a higher pricing tier or add-on will cause 5% of customers to upgrade.
- Not just "better UX," but a 50% increase the success-rate for people moving through the interface.
- Not just "widely used," but 40% of customers surveyed scored at least 4 out of 5 on whether they'd use the feature.

- Not just "customer satisfaction," but moving from a 4/10 to a 8/10 on a survey related to this area of the product.
- Not just "thought leadership," but marketing commits to getting ten external articles to reference it in the next quarter.

Force people to write down exactly what the material change is expected to be. Not because the estimates are accurate, but because it forces the person to think through the answer. Most ideas, we'll eventually admit, are so incremental that we won't be able to measure the effect; that means it is "not material." That's a tough fact to face, but remember the point of the exercise is to force exactly these conclusions, to drastically reduce the field of ideas so that only the actually-best ideas remain. These smaller decisions will make the larger decision easy.

Because this "material change" is just a guess, we won't put it in an equation—no computing with noise! But if you can't justify a magnitude greater than "business as usual," the idea is simply not impactful enough. Your standards are higher than that.

In our example, if we simplistically considered any score that is "6 or greater" to be "material," we'd already have a winner:

Attribute	Player P	Player Q
Health	✓	
Strength		✓
Speed		✓
Endurance		
<b>Total:</b>	<b>1</b>	<b>2</b>

Each "point" in this method is meaningful, so even a difference of 1 point crowns a clear winner. With real-world attributes, and a sufficiently high bar, you will reject nearly all items quickly. People won't like that—their favorite thing will be cut—but it's the only way to stop wasting your time dithering between a pile of things that won't make a difference.

It's also extremely easy to explain your decision: Q materially impacts two important things; everything else is less impactful.



"I haven't yet decided if I'm staunchly in favor, or staunchly opposed. But rest assured, however I vote, it will be staunchly."

It is, of course, still common to have ties. Indeed, if in our example we considered anything "5 or greater" to be material, it's back to a tie<sup>3</sup>:

<sup>3</sup> Although in this case perhaps the problem is that our standards for "materiality" are too low, as opposed to the options being too good.

Attribute	Player P	Player Q
Health	✓	✓
Strength	✓	✓
Speed		✓
Endurance	✓	
<b>Total:</b>	<b>3</b>	<b>3</b>

To address this, we need one more rule.

### Stack-ranked attributes

Back to the top: (1) We're trying to isolate the *one* thing that would be *most* impactful, and (2) complex decisions feel impossible because of a lack of smaller decisions. We've made some decisions already, but we need a few more to isolate the one winner.

We have to decide which attributes are most important. Currently we are treating all attributes as equally important—a check mark next to "Endurance" is equal to a check mark next to "Health," but is that really true?

With a standard rubric, the fact that all attributes are not equally important drove us to reach for "weights." But that computation confounded us with noise. Instead, we simply order attributes by importance, in a single ranked list. No numbers.

"Simply order them" is easy to say but not easy to accomplish, because people get stuck in circular debates:

"Growth is more important than profit, because it's possible to optimize our costs later."

"Yeah, but if we're unprofitable on a unit basis we'll cause a cash-crunch, so we have to be profitable first."

"Yeah, but if it's only about profit, the best thing to do is just 10x prices, and whichever customers stay are super profitable, but that would be wrong."

"Yeah, but if it's only about revenue, the best thing to do is to sell \$1 bills for \$0.80, and that would be wrong."

This conflict highlights the "smaller decisions" that still need to be made.

Both sides are correct in saying it's bad to maximize one thing with no regard to any consequences. But surely you've already ensured your list contains nothing outright absurd<sup>4</sup>. So these *reductio ad absurdum* arguments are moot and can be ignored. Assume (and enforce that) the ideas are sensible, then decide what outcome is most important.

<sup>4</sup> If you have trouble ensuring that items are meeting basic standards, create criteria for an idea making it onto the list to begin with. Examples: it can't be less unit-profitable than some pre-determined target, it can't take longer than N sprints to execute, and it can't require significant retraining of the support team.

A typical ordering for a VC-backed B2B company, optimizing for "growth is paramount, because if growth is there, we'll be able to raise more money," could be:

1. Revenue growth (i.e. “the single biggest driver of equity-value”)
2. Number-of-customers growth (i.e. “market share”)
3. Product experience (i.e. “customers love of the product”)
4. Support cost (i.e. “a cost; more importantly, a measure of usability”)
5. Infrastructure cost (i.e. “a cost; should organically improve with scale”)
6. Net-profit expansion (i.e. “profitable business model”)

A bootstrapped company designed to create wealth for its founders and employees, while being a place where employees genuinely love coming to work and customers genuinely love the product, might be:

1. Cash-basis profit expansion (i.e. “wealth creation” + “mandatory to keep the business alive”)
2. Product experience (i.e. “the reason we get up in the morning is building a great product people love”)
3. Fun (i.e. “I built this business to be the place I *want* to work for”)
4. Number-of-customers growth (i.e. “stagnation is the prelude to death”)
5. Minimizing number of employees (i.e. “we joined a small company to avoid bureaucracy”)

These are generic attributes; “more revenue” could include almost anything. It’s better if your lists are more specific, based on current circumstances, or focused on a subset of the strategy. For example, suppose a product that targets mid-sized restaurant chains is having trouble with customer retention. A better list might be:

1. Dramatic increase in usability  
*The #1 reason customers give us when they cancel in their first year, is that training their employees is too difficult, so the software never gets used.*
2. Reduce costs  
*Price is the #1 reason that long-tenured customers give when they switch to a competitor; reducing costs means we can reduce prices while generating the same profit.*

3. Increase market-differentiation

*If there were features that customers couldn’t get anywhere else, they would stay despite (1) and (2).*

Note how obvious corporate goals like “grow revenue” and “happy customers” are embedded in these goals, but insights or data produce more specific immediate goals. This will cause even better ideas to be selected, and will help the team brainstorm better ideas in the first place.

### **Binstack: The final process**

You’ve finally made all the “small decisions” that make the big decision clear. With your stack-ranked attributes and binary scores of which items materially affect which attributes, here’s what you do:

1. Cross out items that don’t materially address the top-ranked attribute.
2. For the second attribute...
  - a. If no tasks address it, move on to the next attribute.
  - b. If exactly one remaining task addresses it, that’s the winner; you’re finished.
  - c. If multiple remaining tasks affect it, cross out all the others and continue on to the next attribute.
3. Repeat step (2) for the third attribute, fourth, etc.

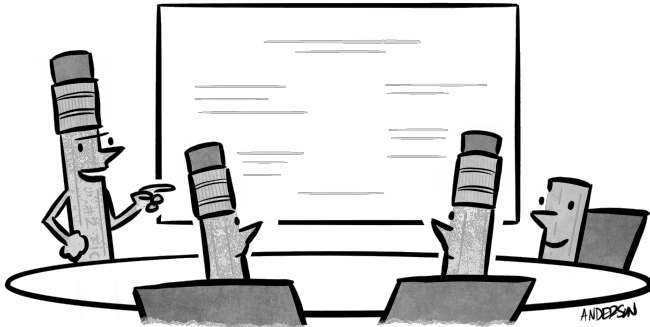
These steps honor our smaller decisions about which results are most important to manifest (ordered attributes), and what these options *really* accomplish for us (binary materiality). It also ensures that we’ll materially affect our #1 attribute; even if another idea moves several other needles, we still have to honor our “small decision” about what is of paramount importance.

The final decision is trivial to explain. It goes something like this:

We decided the most important things we have to accomplish in the next few months are to grow top-line revenue and create defensible technology. Item P does both of these things; none of our other choices did.

Or defending why you didn’t pick some other item:

Item Q is a solid idea; indeed it would both increase profitability and increase our differentiation in the market. However, the most important thing right now is to grow revenue, and item P accomplishes that whereas item Q doesn't. However, in future, if we change our priorities, or complete item P, item Q will be wonderful to consider!



"I like you, Anderson! You're decisive!"

## Real-life errata

### "Effort" conspicuously absent

Many rubrics are set up as "ROI" calculators, i.e. measuring impact relative to the cost of achieving that impact. This often results in selecting less-impactful items, because they are cost-effective to implement. Sometimes that's the right choice, but Binstack is about selecting only for maximum impact, not cost-efficiency.

If you actually want to maximize ROI, use this method. And here's how you decide which method is right for your current decision.

### Fun is underrated. Add it in.

It's still possible to tie. If you have lots of items left over, perhaps your materiality threshold isn't high enough; raise it to thin the herd.

Supposing you have two ideas that are truly indistinguishable, you could flip a coin. I don't recommend that, because you can't explain your choice. A person passionate about the choice you rejected would be upset to hear

you ruled against them so flippantly. Instead, pick whichever item the folks doing the work *want* to do. Do what's fun.

It still sounds flippant. What business does "fun" have in business? When people work on something fun, they work harder and better while enjoying themselves more—more productivity yet more happiness. Do not dismiss this life-hack.

Of course we cannot do what's fun at the expense of what *needs* to be done, but when those two things are not in conflict, why would you not round off in favor of fun?

If you like this idea, take it further: Put "fun" in the attribute stack-rank, and rank it high. Even second position is not ridiculous. Knock that #1 priority out of the park while having fun. What's wrong with that?

### Evolving stack-rankings, and different stack-rankings per team

You should expect the stack-ranking to change over time, even rapidly. In the early days of a company, just getting any customers is hard, at any price, so that might be much more important than revenue or profit. A mature company who reliably gets customers in the door might be more interested in expanding efficiency or profitability, not because growth is unimportant, but because it's so systematic that it is no longer an existential threat, and other things are more pressing.

If you have multiple teams, and therefore the time to execute multiple items, you might want separate lists for different goals. For example, you might say, "We want one initiative that will materially increase profit, one that materially increases our internal effectiveness or efficiency, and all the rest should maximize growth." Each of those would be the number-one item in its own stack-ranked list, and the attributes below that might be copied from the company-wide general list.

In all cases, remember to hone the attributes with more specificity, to generate better ideas.

## What if nothing is left?

“*I’ve generally found that the best product ideas live at the intersection of “duh” and “holy shit.”*”

—[Aaron Levie](#)

Sometimes we’re so harsh with our materiality threshold that none of our ideas meet our exacting standards. What does that mean?

It means your ideas aren’t good enough. It means your problem wasn’t one of prioritization after all, but rather of not having ideas worth prioritizing.

Focus the team on this new, more pressing problem: To generate better ideas. [Here’s some help with generating better ideas.](#)

But this will take time! Fine, put it in the sprint. But we need a plan right now! Too bad; it’s better to take a month to find a wonderful thing to spend the rest of the year on, than to plod along doing things that aren’t valuable enough. Do [high-ROI](#) small projects in the meantime.

## Why bother scoring everything when most will be rejected immediately?

Indeed, you needn’t bother. That saves time.

However, remember that an important aspect of decision-making is explaining your decision. Often, explaining “why not Q” is just as important as explaining “why P” Other people will want to know that you seriously considered other options.

Most importantly, there are people who really wish you had selected Q. Maybe Q was even their idea. The rejection will be easier to accept if Q was seriously and genuinely considered. Perhaps, by being invited into the decision process, the person who invented Q will come to

the right conclusion on their own. This is important; do not underestimate the human—and humane—part of the process.

## Reductive

Binstack can feel reductive—over-simplified, ignoring the reality of a complex world, therefore resulting in an incorrect conclusion. It is true that for complex problems like foreign policy, national economics, and climate change, a reductive approach is invalid.

But for finding the right feature to build, or the right marketing campaign to launch, or the right bug-tracking software to adopt, or the right database to use for a new project, or the best candidate to hire, being (intelligently) reductive is how you transcend the noise, arrive at a clear decision, and explain it to others.

This isn’t foreign policy, it’s a feature list. Make an impact with Binstack, and be happy!

---

Printed from: *A Smart Bear*

<https://longform.asmartbear.com/maximized-decision/>

© 2007-2024 Jason Cohen  @asmartbear