

The BlueCore™ Solution Model

The BlueCore™ Solution Model

At Blue Margin, we've developed an RCA model ("BlueCore™") for designing data warehouses, dashboards, and our internal strategies and systems. The model is nothing new under the sun. It employs established RCA principles, and it works because it fits how we think.

Using BlueCore™ successfully requires a balance of structure and flexibility. Following your gut alone will cause your goals to quickly fade in the mist as you glom onto design elements. Make the structure too ridged, and you'll get bogged down trying to tie every loose end. Feel free to borrow what works and leave or modify the rest, just be intentional about establishing a standard model that can be easily duplicated throughout your organization. Through trial and error, and by incorporating what we found to be the most effective elements in other models, this approach consistently results in solutions that *get adopted*, both internally and for our clients.



Tips for a Successful BlueCore™ Analysis:

Keep your eye on the prize. Throughout the process, remember the purpose of the BlueCore™ model: to identify the simplest solution that most efficiently advances your goal. In our experience, intuition-driven solutions are almost always tragically flawed. They map too many features to too many issues, resulting in inefficient solutions that don't get adopted. We have found that using a framework that organizes a problem into fewer moving parts leads to a more durable solution.

Keep it loose. This is a flexible framework, not a rigid formula. The trick is to stay centered on the goal you're trying to achieve. If your goal is clearly defined, the high-level process is simple:

- 1) Identify the problems hindering your goal
- 2) Drill down until you get to the *actionable* root causes underlying those problems
- 3) Design your solution to address those root issues

Feel free to “bend the rules.” If someone jumps to design, don't hit the buzzer. Just make a note of their idea, then carry on with the process. If your “root issues” crisscross, overlap, or merge, don't worry about the mess it makes. Just keep drilling down to root causes, and the picture will come into focus. Also, don't forget to occasionally step back and make sure the overlying business goal is still in focus.

Avoid jumping to design. One of the reasons new systems fail is the human tendency to leap from problem to solution. For example, if someone at your company sounds an alarm (“We need to reduce customer churn!”) the responses will likely center on design (“We should implement better customer-service software!”) Sounds reasonable. It may seem like the self-evident solution to the problem. But there's invariably more to the story. You'll find the rest of the story somewhere underneath the acute and apparently all-important symptom of *customer-churn*. The rest of the story is revealed when you thoroughly explore the issues *before* exploring possible solutions.

Even after you start digging, each unearthed issue will likely re-trigger the impulse to get busy applying fixes. Solution design is more about defining the goal and the issues hindering that goal than about defining features and functionality. Do the first part right and the second part will reveal itself. Again, if compelling design ideas crop up, you can always put down your shovel, make a note, then get back to digging.

Use the 80/20 filter. The brainstorming process needs to remain fluid. If you become fixated on perfection or try to evaluate the merit of every idea along the way, the process will bog down. You'll find yourself trapped in a binary, pass/fail framework, instead of prioritizing issues on a scale and simply limiting yourself to the most critical ones.

So, feel free to write down any and every issue. Let the ideas flow unhindered, but then apply a filter to separate the must-haves from the nice-to-haves. Ranking each issue from one to five (five being the most critical), then dismissing the ones and twos (and possibly threes) will help you overcome the perfectionism

trap. If you struggle to let go of lower-priority issues, console yourself with the idea that once your solution is established, you can go back and address those lesser issues (though in our experience, they'll never qualify as worth fixing). The process of deciding what's above-the-line and what's below-the-line is central to developing solutions that work. Try to fix everything and you'll never arrive at a solution that's simple enough to endure when transferred from the white board to the real world.

Chapter 34: The 5 BlueCore Steps:

Overview. In its simplest form, the BlueCore™ process starts with a measurable goal, then lists the successive issues hindering that goal until actionable roots are identified, revealing the simplest, highest-impact solution.

1) Define the Goal (in measurable terms)

Start by defining what you're after. This sounds easy, but takes some effort. Arriving at a well-stated goal that's "at the right level" can be tricky.

Don't shoot for a goal that's too lofty, nor one that's too tactical. To illustrate, "Improve operational efficiency" is probably too high-level and vague to be useful. On the other hand, "Create a better expense reimbursement form," sounds like a solution in search of a goal.

A better goal might be, "Decrease expense-reimbursement turnaround from five days to three." This goal is fairly specific, but doesn't dictate a particular design. It also identifies a measurable KPI (i.e., "Average days to process expense reports"). Defining an empirical measure of success is central to designing a successful solution.

2) List the Problems

After the goal has been identified, brainstorm all the problems that most directly hinder the goal. An example problem might be, "Employees don't fill out expense reports correctly." It's a simple statement that doesn't necessarily explain "why" (answering "why" comes next).

After you've listed the problems, rank them 1-5 to determine which are above or below the line (our scoring methodology is described below).

3) Identify the Proximate Issues and Root Causes

After you've determined which problems are above-the-line, start with the most important and drill down to proximate issues and root causes for that problem. For example, a

BlueCore™ Discovery & Design Model



GOALS

- Your highest-return goals drive the BlueCore™ analysis and design process.
- Prioritization is key, highest-value ("above-the-bar") goals only.



PROBLEMS

- List all problems, no filtering.
- Intentionally call out "insurmountable" problems.



CAUSES

- Drill down from proximate to root causes.
- Be willing to "touch the nerve" - no sacred cows.



DESIGN

- Consider multiple designs; assess possible sub-optimal outcomes.
- Test designs conceptually before deploying.



EXECUTION / EVOLUTION

- Agile POC: Define the solution. Seed adoption with team of project "champions".
- Deploy & Support. Hands-on training, on-time support.
- Auto-output of KPI's, Periodic Review, CPI.

proximate issue for the above example might be, “The expense form is frustrating and confusing to employees.” Ultimately, you’re trying to drill down to the root cause of the problem, which might take digging deeper. You can get to the root by repeatedly asking *why*. For example, the next level of causation might be that the expense form has redundant fields, or ambiguous wording, or doesn’t account for edge-cases.

By continuing the process, you’ll drill down to root causes. In our experience, these typically show up three to five layers deep. For example, a root cause might be, “The expense form was designed from a bookkeeping perspective and doesn’t take end-users’ viewpoints into consideration.” Or, if you dig deeper, you may decide the real root is, “There is no system in place for periodically reviewing company forms for ease-of-use and applicability.” Either root might be at the right level to address. You get to decide. Here is where some flexibility and intuition come into play.

Take your time to get to the root. In the above example, it might be tempting to take action at the “problem” level by issuing a new policy penalizing employees for incorrectly filling out expense forms (“Expense forms filled out incorrectly will be returned and ineligible for processing until the next cycle.”). Rather than applying band-aids, drilling to roots will point to a more foundational solution that doesn’t just address the symptoms, for example, “Redesign the expense form to fit the end-user’s perspectives and workflows.”. By getting to root causes you’ll also fix, by extension, the overlying proximate issues or symptoms. The result will be greater adoption and better outcomes.

Rather than applying band-aids, drilling to roots will point to a more foundational solution that doesn’t just address the symptoms.

Once you’ve drilled down to the root cause(s) for a problem, move to the next most important problem and repeat the process. Each problem might have several proximate issues and root causes, which can seem daunting. However, problems often share underlying elements, making the process easier as you go along, similar to how a puzzle becomes progressively simpler with each added piece.

4) Assess Potential Designs

Once the roots are identified, the design should be fairly self-evident. You can evaluate the efficacy of potential designs by simply counting how many overlying causes each solves. Additionally, considering possible unintended consequences of each design will help you determine which will introduce the least risk.

The most important litmus test for the viability of a possible solution is assessing how efficiently it will advance the KPI(s) defined in the Goal. Also, consider how difficult it will be to track those KPIs. Is the performance data a byproduct of normal workflows, or does it have to be separately entered? If the data is already in a transactional system, how accurate is it, and how difficult to retrieve? At Blue Margin, we won't adopt a solution unless it's worth putting on a dashboard. So, if the cost of acquiring the data to measure progress is not outweighed by the value of solving the problem, the proposed solution doesn’t make the cut.

5) Execute and Evolve

Keep in mind that that every design needs room to evolve. Even after carefully following the BlueCore process, your solution will require iteration before it’s running at top-speed. It will also need periodic review to adapt

to your evolving business. Keeping KPIs front-and-center will naturally drive that evolution. In other words, if you show employees empirically (and in pictures) how they're performing against goals, you'll harness the strategic problem-solver in each of them, and they'll naturally adjust as they go.

That said, minimizing course corrections before you launch will greatly increase the likelihood of adoption. End-users will become quickly jaded and will reject new systems if they encounter too many hurdles out of the gate.

BlueCore™ and Dashboard Development

Since adopting the BlueCore model, we're consistently amazed at the number of sophisticated businesses who suffer with poorly adopted reports and dashboards that were designed by intuition, rather than a structured process.

Typically, companies falter at the very first step: defining the goal. This mistake sets a bad trajectory for the rest of the design process, leading to an accumulation of features and functionality that poorly support the goal.

Typically, companies falter at the very first step: defining the goal.

To define the goal(s), ask yourself the following questions:

- 1) What are the company's top three opportunities? In other words, what is most exciting about the business currently?
- 2) What are the top three business problems in the company? In other words, what keeps executives up at night?
- 3) What are the most critical activities, by role, that affect the company's success?
- 4) What crucial decisions are made on a daily/monthly/quarterly basis that best differentiate the business and make it successful?
- 5) How do executives, the management team, and employees maintain focus on the right decisions and activities?

Execution comes down to decisions and actions, so focusing on which decisions and actions are most important is the right place to start when thinking about developing dashboards. It can be tempting to create a laundry list of metrics and reports that sound like the typical information a company should care about, or that include the same data as the reports you currently use. Instead, we recommend setting those ideas aside and thinking about how you measure your company's success on a *daily* basis.

Then, simply follow the BlueCore™ (or another RCA) model. Ask yourself what most hinders a given goal. If the goal is to double sales, fearlessly call out every issue that hinders that goal, without regard to how significant the issue is, or how seemingly insurmountable. Don't be afraid to invite the elephants into the room. If the obstacle is daunting (i.e., "We've tried everything we can think of and are out of ideas"), continue working your way through the process and you'll be surprised at how many viable solutions present themselves. For example, the answer to the above issue might be to bring in a consultant.

Once you’ve drilled down to root causes (e.g., “We don’t accurately incentivize our salespeople to focus on the right activities.”), you can begin to design your dashboard. This is where KPIs do their magic. Keep in mind the axiom *what gets measured, gets managed*. If you design a solution but have no means of measuring its effectiveness, you’ll be back at the same table addressing the same issue soon.

If you design a solution, but have no means of measuring its effectiveness, you’ll be back at the same table addressing the same issue soon.

For KPIs to have value, they must be high-impact. In other words, creating countless spreadsheets, charts, graphs, and PowerPoints is not the key to changing behavior. The key is to find those few measurements that really touch a nerve and drive performance. To find them, consider which metrics measure the activities that produce results, rather than those that measure the results themselves.

For example, “Closed Sales Per Month in Dollars” is a precise measurement of outcomes, but to change behavior, you should put a spotlight on the activities that produce those outcomes. Choosing KPIs that measure downstream results such as Closed Sales can be difficult for employees to take ownership of, which will limit motivation.



Blue Margin dashboard with KPIs. Source: Blue Margin Inc. Internal Graphic

In our experience, KPIs should measure activities that are as close in proximity to the desired result as possible, while remaining largely under employees' control. "Number of Dials" might seem like a good sales KPI because the employee has near-perfect control over it. However, depending on the business, *dials* might be the farthest point in the sales process from a closed deal, and therefore may be a poor indicator of performance.

Additionally, measurements like *dials* are high in volume and are therefore difficult to audit. Validating each dial is impractical, making it easy to game, even if subconsciously. After all, what constitutes a dial? Do misdials or wrong numbers count? How about leaving a message? What about leaving a message at the same number twice in one day? Moreover, if KPIs are tied to activities that predict outcomes poorly (i.e., Dials), employees will have little motivation to push for the end-goal (i.e., Closed Sales).

You might argue that placing careful parameters on what constitutes a dial would solve the problem. However, the more complicated the "rules," the higher likelihood of confusion and poor adoption. Alternatively, you might argue that additional KPIs could be added to motivate the employee to advance the sale, but adding too many KPIs can be equally problematic, creating complexities that cross the threshold of diminishing returns. For visibility to take hold, simplicity is essential.

For visibility to take hold, simplicity is essential.

Going to the other extreme, one might decide to measure the number of proposals generated per salesperson, an activity with a high correlation to sales outcomes. The problem with this level of measurement is that the employee may have little control over the results on a day-to-day basis. The likely result is that employees will be discouraged rather than motivated by the pressure to perform, because their ability to impact the KPI is too limited.

Instead of "Proposals Generated," a good KPI might be "Sales Conversations." Each Conversation would represent an exchange between the salesperson and a prospective client that moves the process along (one way or the other). *Sales Conversations* is an activity reasonably within the employee's control, yet with a strong correlation to closed sales. Moreover, KPIs such as "Sales Conversations" are auditable. That is, they are few enough in number (maybe three to five per day) and have defined attributes (e.g., With whom? When? What was discussed?) such that a manager can review and qualify the data behind the KPI.

Ultimately, identifying the right KPIs often is the solution itself.

Ultimately, identifying the right KPIs is often the solution itself. In other words, rather than implementing new systems or processes, simply providing visibility into the right metrics will drive the right activities.