The Mid-Size Data™ Playbook

How start-up founders and CMOs can use mid-size data techniques to prioritize marketing tests, read small-data results honestly, and automate the next steps

---

The Mid-Size Data® Approach
Research review · Framework · Playbook

ENGINEERED BY
ROI FACTORY

Executive summary

Giant marketers have the luxury of a testing superhighway due to their high traffic volumes. Booking.com, Microsoft and Netflix run hundreds or thousands of experiments at once, so a slow or failed test costs them almost nothing. A start-up or a growing business usually has just a single lane: enough traffic, budget and attention for one meaningful test at a time. Every week a bad test sits in that lane is a week the next good idea waits.

We reviewed the published research and platform documentation on test prioritization, small-sample statistics, enterprise experimentation programs and test automation. Our findings:

FREE

Free Mid-Size Data™ Test Planner

We did the research, published the whitepaper, and now we've built the Planner. It's free, no email or signup required.

Try it out!

1. **There is no established testing hierarchy built for start-ups or small and medium businesses.** The well-known scoring models (ICE, PIE, PXL, RICE) rank ideas by impact, confidence and ease. None of them asks the question that matters most with one lane: *can our traffic even detect this change, and how many weeks of the lane will it use?*
2. **What your traffic can detect is set by the number of leads or sales in the test, not the number of visitors.** With about 100 leads in the whole test, a standard test can only reliably detect a lift of roughly 50% or more. Very few changes can ever move results that much.
3. **Most ideas do not win, even at the best-run companies.** Microsoft has reported that about one third of tested ideas improved the metric they targeted. A meta-analysis of 1,001 tests run by experienced practitioners found a median lift of 0.08%. Small tweaks are mostly invisible to small-data tests.
4. **You can read results faster without fooling yourself.** Sequential tests let you check results on a schedule and stop early when the answer is clear; one real-world analysis found they used about 26% less traffic on average than fixed-length tests. Bayesian decision rules, variance reduction, earlier proxy measures and choosing a confidence level that fits the risk all shorten the wait. Checking results every day and stopping on the first good number does not: in one well-known example it raises the false-alarm rate from 5% to 26%.
5. **Enterprise testing teaches discipline more than tools.** The lessons that transfer are: expect most ideas to fail, write the decision rule before launch, check that the data can be trusted, distrust surprising wins, and let results overrule opinions, including the boss's.
   Startups that adopted A/B testing performed 30% to 100% better after a year in one large study.
6. **Testing can be partly automated today.** Google Ads applies favorable experiment results automatically by default for some experiment types. Testing tools offer "bandit" modes that shift traffic toward the leader on their own. The safe pattern is automation inside written rules, with a person checking anything surprising or hard to undo.

From this research we built the Mid-Size Data™ approach; a seven-step way to choose, run, read and automate tests when you can only run one at a time. It ranks ideas by *expected value per week of lane time*, sorts out ideas that do not need a test at all, and climbs a **testing ladder** from big swings (offer, audience, message) down to small details only as traffic grows.

1. The problem: one lane, slow answers

Why small teams can't just test more

A test needs enough people to tell a real difference from luck. Big sites get that in days or even hours, so they can split traffic many ways and run many tests side by side. A business with a few thousand visitors a month gets it in weeks or months, and splitting that traffic across two tests at once makes both slower. In practice, small teams run one test at a time, often on one page or one ad campaign.

That turns test selection into the main decision. The true cost of a test is not the tool or the design work. It is the lane time: the weeks of traffic it uses up, and the better idea that waits behind it. Google statistician Steven Scott makes the same point about online experiments generally: the cost of experimenting is "dominated by opportunity cost rather than the cost of obtaining experimental units" [17].

Most ideas don't win

Teams usually overestimate how often their ideas will work. The published numbers from companies that test at scale are sobering:

- At Microsoft, Ronny Kohavi and colleagues reported that about one third of tested ideas improved the metric they were designed to improve, one third made no measurable difference, and one third made things worse [1].
- In a 2022 paper, Kohavi, Alex Deng and Lukas Vermeer presented success rates of 15% for Bing, about 10% for Booking.com, Google Ads and Netflix, and 8% for Airbnb Search [2].
- Harvard Business School professor Stefan Thomke, who studied Booking.com and others for his book *Experimentation Works*, says that on average only about 10% of experiments improve the bottom line [3].
- Georgi Georgiev's analysis of 1,001 tests run on his platform, mostly by experienced practitioners, found that 33.5% produced a statistically significant win. But the median estimated lift across all tests was just 0.08%, and the mean was 2.08% [4].

Two lessons follow. First, a backlog full of small tweaks will mostly produce "no clear winner." Second, when the lane is scarce, the ideas you choose to test matter more than how well you run each test.

The small-data trap

Small teams fall into a predictable pattern:

1. Test a modest change (a button, a headline, an image).
2. Watch the dashboard daily.
3. Stop when it shows "95% chance to win," or give up after a few weeks of "no clear winner."
4. Ship the "winner," see no change in the business, and lose faith in testing.

Each step has a known cause and a known fix. The rest of this paper covers both.

2. What your traffic can and can't see

Before ranking ideas, find out what your traffic can detect. Statisticians call this the **minimum detectable effect**: the smallest real lift a test of a given size will reliably catch (by convention, 80% of the time at 95% confidence).

The table below shows it for a four-week test split 50/50 between the current version and one new version. We computed it with the standard two-group sample-size formula (see Appendix A). It is an estimate for planning, not a guarantee.

Smallest lift a four-week test can reliably detect

Visitors in 4 weeks	Lead rate 2% (leads in test)	Smallest detectable lift	Lead rate 5% (leads in test)	Smallest detectable lift
2,000	40	~88%	100	~55%
5,000	100	~55%	250	~35%
10,000	200	~39%	500	~24%
25,000	500	~25%	1,250	~15%
50,000	1,000	~18%	2,500	~11%
100,000	2,000	~12%	5,000	~8%

Assumptions: 95% confidence, 80% power, two-sided test, 50/50 split, one variant against the current version. "Lift" is relative: a 50% lift takes a 2% lead rate to 3%.

Three things stand out.

It's the count of leads or sales that matters. Read across the rows: 5,000 visitors at a 5% lead rate and 10,000 visitors at a 2.5% rate both give about 250 leads and roughly the same detectable lift. A useful rule of thumb: with about 100 leads or sales in the whole test, only lifts of about 50% or more show up reliably; with about 1,000, lifts of about 18%.

Detecting smaller effects gets expensive fast. The traffic you need grows with the square of how small an effect you want to see. Halving the detectable lift takes four times the traffic. Evan Miller's well-known rule of thumb puts it simply: the sample needed per group is about 16 × variance ÷ (smallest effect)² [5].

A test that can't detect the likely effect is not a test. If your idea might plausibly lift leads by 10% and your lane can only see 40%, the test will almost certainly end with "no clear winner," however good the idea is. Kohavi's advice is blunt: if the sample-size calculation shows you need more traffic than you get in a typical two-to-six-week window, question whether the test is worth running [11].

There is a second, sneakier problem. When a test that is too small *does* show a significant win, the measured lift is usually exaggerated. Kohavi notes that "the lower the power, the more exaggerated the effect may be" [11]. A small test that "wins" by 40% may really be a 5% improvement, or nothing at all.

3. Is there an established framework for small businesses?

Short answer: there are well-known frameworks for scoring test ideas, and one well-known framework for choosing channels, but none is built for a single lane with mid-size data. Here is what exists.

Scoring frameworks

Framework	Origin	How it scores	Strength	Gap for a single lane
ICE (Impact, Confidence, Ease)	Popularized by Sean Ellis, GrowthHackers [15]	Score each 1–10 and combine	Fast, easy to start	Impact and confidence are guesses; easy to inflate a pet idea [15]
PIE (Potential, Importance, Ease)	Chris Goward, WiderFunnel [22]	Score each and combine	Weights pages by traffic and value	"Potential" is subjective [15]
PXL	Peep Laja, CXL (2016) [15]	Mostly yes/no questions: above the fold? noticeable in 5 seconds? adds or removes something? high-traffic page? backed by user testing, surveys, heat maps or analytics? plus effort bands by time	Rewards evidence over opinion; harder to game	Built for website conversion tests; ignores whether the traffic can detect the change
RICE (Reach, Impact, Confidence, Effort)	Sean McBride, Intercom (2018) [20]	(Reach × Impact × Confidence) ÷ Effort; impact on a 0.25–3 scale; confidence 100/80/50%	Forces reach in real numbers; penalizes effort	Built for product roadmaps, not tests; effort is team time, not traffic time

CXL's own critique of ICE captures the shared weakness: "if I could guess what the impact would be, why would I even test it?" [15]. PXL answers by asking for evidence. Still, all four treat a test's cost as the work to build it. With one lane, the bigger cost is the weeks of traffic it consumes, and none of the four measures that.

Channel testing for start-ups

Gabriel Weinberg and Justin Mares's *Traction* introduced the **Bullseye Framework**: list all 19 ways to win customers, pick the few most promising, run cheap tests on them, and focus on the one that works [21]. It is the closest thing to a mid-size data testing hierarchy, but it answers "which channel?" rather than "which test next, and how do we read it?"

Tempo

Sean Ellis's "high tempo testing" set a goal of launching three tests per week, with a weekly growth meeting to review results and pick the next ones [23]. Tempo is the right instinct. But three tests a week assumes enough traffic to read three tests a week. With one lane, tempo comes from choosing tests that resolve quickly, not from launching more of them.

Sequencing by value per unit of time

Product development offers a better model. Don Reinertsen's **cost of delay divided by duration** (CD3, also called Weighted Shortest Job First) ranks work by the value lost per week of waiting, divided by how long the job takes [19]. Short, valuable jobs go first; many short wins can beat one long, larger one. That is exactly the trade-off a single testing lane faces. We borrow it below.

The hierarchy of evidence

Medicine ranks evidence from expert opinion (weakest) through observational studies to randomized trials (strongest). Kohavi applies the same idea to online business: controlled experiments are the strongest proof of cause and effect, and single observational analyses deserve skepticism [25]. The useful lesson for small teams is that not every decision needs the top of the pyramid. Cheap, reversible decisions can rest on weaker evidence. Expensive, hard-to-undo ones deserve a real test.

Our verdict

The pieces exist: evidence-based scoring (PXL), value per unit of time (CD3), channel selection (Bullseye), and a hierarchy of evidence. What is missing is a method that joins them around the one constraint small teams actually have.

So we built it! And we made it free.

---

4. The Mid-Size Data™ Method

This method is ROI Factory's synthesis of the research above. It is not an industry standard. We use it with clients and refine it as results come in.

Step 0: Make sure the numbers can be trusted

Testing on broken tracking wastes the lane. Before the first test:

1. Check that every lead and sale is recorded once, with the source it came from.
2. Check that the testing tool splits traffic the way you set it. Kohavi reports that a broken split (called sample ratio mismatch) happens in about 6–10% of all A/B tests and invalidates the results [11].
3. Where you can, run an A/A test: the same page against itself. If it "finds a winner," something is wrong with the setup [11].

Step 1: Sort ideas into Decide, Research, Test or Watch

Not every idea deserves the lane.

Bucket	When	What to do
Decide	Low risk, easy to reverse, or clearly better (a broken form, a missing phone number, a confusing price)	Just make the change and keep an eye on the numbers
Research	You don't know *what* to test yet	Run cheaper research first: customer interviews, surveys, session recordings. PXL gives extra weight to ideas backed by this kind of evidence [15]
Test	Meaningful upside, real uncertainty, and the lane can detect the likely effect	Score it (Step 3) and queue it
Watch	Worth doing but the likely effect is far below what the lane can see	Bundle it into a bigger change, move it to a place with more traffic (often ads), or ship it as a Decide item if the risk is low

Step 2: Size the lane

Use the table in Section 2, or our free ROI Factory Mid-Size Data™ Testing Planner: *with our traffic, what is the smallest lift we can detect in two, four and eight weeks?* Write the answer down. It becomes the bar every idea must clear.

Step 3: Score by value per lane-week

For each Test idea, estimate:

- **Reach:** how many leads or sales a month pass through the thing you're changing.
- **Lift if it works:** a realistic guess, not a hope. Remember the enterprise numbers: most real effects are small.
- **Chance it works:** start at 20–30% for a typical idea (consistent with the success rates in Section 1) and raise it only for evidence: research findings, a competitor's proven approach, a past win. A PXL-style yes/no checklist keeps this honest.
- **Lane-weeks:** how many weeks of traffic the test needs to detect that lift, plus any build time that blocks the lane.

Then:

> **Priority score = (Reach × Lift if it works × Chance it works) ÷ Lane-weeks**

Two gates apply before scoring:

1. **Detectability gate:** if the realistic lift is below what the lane can detect in eight weeks, the idea goes back to Watch.
2. **Downside gate:** if the change could hurt the brand, pricing or legal position, it needs a person's sign-off before it enters the queue.

Step 4: Climb the testing ladder

Effect sizes tend to shrink as you move from *what you sell and to whom* down to *how a page looks*. Mid-size data teams should start at the top, where effects are big enough to see, and only move down as traffic grows.

Rung	What you test	Why it's here
1. **Measurement**	Lead and sale tracking, the split itself	Nothing else is readable until this works
2. **Offer**	What you offer, how it's priced and packaged, the guarantee, the free first step	Changes the reason to act; the largest effects usually live here
3. **Audience and message**	Who you target, the problem you lead with, your main promise	Changes who shows up and why
4. **Channel**	Which ad platform, search vs. social, partners, email	Bullseye-style tests of where buyers are [21]
5. **Page and path**	Page structure, number of steps, form length, proof placement	Real but usually smaller effects
6. **Creative**	Images, video, ad variations, headlines within the same message	Best tested inside ad platforms, which have more traffic and built-in tools
7. **Details**	Button color, small copy edits, layout tweaks	Usually only detectable with large traffic; mostly Decide or Watch items for small teams

The ladder is a default, not a law. A rung-7 fix that removes a real obstacle (a form that fails on phones) can beat a rung-2 idea. That is what the Decide bucket is for.

Step 5: Write the decision rule before launch

> ***This step does more for test quality than any tool. It prevents the most common errors: moving the goalposts, peeking, and arguing about the result afterward.***

Before the test starts, write one page that says:

1. **The hypothesis:** "Because [evidence], we believe [change] will [effect] for [audience]."
2. **The one main measure** that decides the test, and two or three **guardrail measures** that must not get worse (for example, lead quality or cost per lead). Large testing programs call the main measure the Overall Evaluation Criterion [26].
3. **The smallest lift worth acting on** and the **confidence level**, chosen for the risk (Section 5).
4. **When you'll look, and the stopping rules:** stop early for a clear win, stop early for clear harm, stop for "no meaningful difference likely," and the maximum run time.
5. **What you'll do for each outcome**, including "no clear winner."

Step 6: Read fast, honestly

Use the methods in Section 5 (below) to get a valid answer in the fewest lane-weeks.

Step 7: Log, learn, re-score

Record every test: hypothesis, result, lift estimate with its range, and what you did. Re-score the backlog after each result. A win on rung 3 (a new message) often creates new rung-5 and rung-6 ideas; a loss lowers the "chance it works" for similar ideas. Over time the log becomes your own benchmark for realistic lifts, which makes Step 3 more accurate.

---

5. Reading valid results quickly with mid-size data

The goal is the shortest honest wait. Here are the methods, in rough order of how much they help small teams.

5.1 Stop peeking without a plan

Most testing dashboards show a live "chance to beat the original." The math behind that number assumes you look once, at a sample size you fixed in advance. Look every day and stop at the first good number, and the error rate balloons. In Evan Miller's example (a test checked after every visitor, stopped at the first "significant" result or after 150 visitors), a change that does nothing gets declared significant 26.1% of the time instead of 5% [5]. Even ten planned looks need a much stricter bar: to keep a true 5% error rate, each look must reach about 1% [5]. Spotify's engineers show the same inflation in their own simulations [6].

You don't have to stop looking. You have to look in a way the math accounts for.

5.2 Use a sequential test

Sequential tests are designed for checking results on a schedule. They set stricter bars at early looks and relax them later, so you can stop early when the answer is clear without inflating errors.

- **Group sequential tests** (from clinical trials) check at planned points, such as once a week. Spotify uses them and found in simulations that they gave the most power when you can estimate the maximum sample size in advance [6].
- **"Always valid" tests** let you check at any time. Optimizely's testing engine is built on this approach [7]. Spotify found them a good choice when you can't estimate the sample size, with somewhat less power [6].
- **The payoff:** in Georgiev's 1,001-test analysis, sequential tests stopped at about 64% of their planned maximum run time on average and used about 26% less traffic than equivalent fixed-length tests [4].

Sequential tests can also stop for **futility**: when it becomes very unlikely the change will reach the lift you care about. For a single lane, stopping a dead-end test early is as valuable as stopping a winner early, because it frees the lane for the next idea.

For small teams: pick a tool that offers sequential testing, plan a weekly check, and set a maximum run time (for example eight weeks).

5.3 Use a Bayesian decision rule, carefully

Bayesian methods report what decision-makers want to know: "there's a 92% chance B is better, and if we're wrong, we expect to lose very little." Two ideas make them useful for small teams:

- **Expected loss and a "threshold of caring."** Chris Stucchio's approach, used in VWO's testing engine, stops a test when the expected loss from picking the wrong version falls below a level too small to matter to the business [16]. This turns the question from "are we sure?" into "is the remaining risk worth more lane time?" That is the right question for one lane.
- **Priors from past results.** A prior is what you already know about typical effects. Google's Conversion Lift studies now use Bayesian methods that draw on historical campaign data, which Google says lets advertisers run studies "with lower budgets and fewer conversions" [13]. GrowthBook's optional prior shrinks small-sample results toward zero so you "do not over-interpret small sample sizes" [14].

The caution: Bayesian results are not immune to peeking. GrowthBook's own documentation notes that stopping early "can still result in inflated false positive rates" [14]. Kohavi, Deng and Vermeer also warn that some "seemingly intuitive concepts" promoted by testing vendors and agencies are "misleading, often badly so" [2]. Use a written stopping rule either way.

5.4 Choose the confidence level for the risk

95% confidence is a convention, not a law. Georgiev found that experienced practitioners chose thresholds between 80% and 95% to match the stakes, and called the one-size-fits-all 95% "typically suboptimal from a business perspective" [4]. Google Ads experiment reports default to an 80% confidence interval; Google recommends running experiments two to three weeks and lets you pick other levels [12]. Google's Conversion Lift treats results below 90% as directional [13].

A practical rule:

Change	Suggested confidence	Why
Easy to reverse, low downside (ad creative, a headline)	80–90%	Being wrong costs little; speed matters more
Moderate (page structure, offer framing)	90–95%	Balanced
Hard to reverse or high stakes (pricing, brand, legal claims)	95%+ and a repeat test	Being wrong is expensive
Surprising, large win	Stricter bar or a repeat	Twyman's law (below)

Lower confidence buys speed at the price of more false wins. Make that trade on purpose and write it in the decision rule.

5.5 Reduce the noise

Variance reduction (CUPED). Microsoft's Alex Deng and colleagues showed that adjusting results using each person's behavior *before* the test cut noise by about 50% at Bing, the same as doubling traffic [9]. Several testing tools now offer it. It needs pre-test data on the same people, so it helps most for returning customers, logged-in users or email lists, and little for brand-new visitors.

Test where the traffic is. Ad platforms randomize huge audiences and offer built-in tests (Google Ads Experiments [12], Meta's A/B testing in Experiments). ***A message or offer test often resolves faster in ads than on a low-traffic website, and the winner can then move to the site.***

Bundle small changes. If three related changes each might lift results 5%, test them together as one bolder version. You learn less about which change did it, but you get an answer.

5.6 Measure something earlier

When the thing you care about (closed deals) takes months, decide on an earlier step that predicts it: a qualified lead, a booked call, a trial that reaches a key action. Stanford and Harvard economists Susan Athey, Raj Chetty, Guido Imbens and Hyunseung Kang formalized this as a **surrogate index**: combine several short-term measures into a prediction of the long-term outcome. In their job-training example, six quarters of data estimated nine-year effects with 35% smaller error bars [10].

Two rules keep earlier measures honest. First, check against your own history that the early measure actually predicts the later one. Second, keep the later outcome as a guardrail and look at it when it arrives. A test that wins on form fills but loses on lead quality is a loss.

5.7 Distrust surprising wins

Large testing programs follow **Twyman's law**: any figure that looks interesting or different is usually wrong [11]. A 60% lift from a headline change is more likely a tracking error, a bot surge, or a small-sample fluke than a breakthrough. Kohavi, Deng and Vermeer show that the lower a program's success rate, the more of its "wins" are false [2]. For small teams this means: check the data, check the split, and repeat any surprising win before you build on it.

6. What enterprise testing teaches

Lessons that transfer

1. **Expect most ideas to fail.** Plan the backlog and the budget around a one-in-three-to-one-in-ten hit rate [1][2][3]. That is normal, not a sign the program is broken.
2. **Write the decision rule first.** Pre-set measures, stopping rules and actions are what make results trustworthy (Section 4, Step 5).
3. **Pick one main measure, plus guardrails.** Large programs decide on one Overall Evaluation Criterion that predicts long-term value, and watch guardrail measures that must not get worse [26].
4. **Trust checks are routine.** Split checks, A/A tests and data-quality alerts run on every test [11]. GrowthBook, for example, automatically checks for broken splits, too little data, and suspiciously large lifts [14].
5. **Use sequential tests and variance reduction.** Spotify, Optimizely and Microsoft built these into their platforms because they shorten tests without raising error rates [6][7][9]. Mid-size data teams can now get both in off-the-shelf tools.
6. **Let results overrule opinions.** Thomke tells of a new Booking.com CEO who presented a new logo. Employees replied that they would check it with an experiment, and the experiment decided whether the logo could stay [3]. At Booking.com, 75% of core employees can run experiments [3].
7. **It pays off for start-ups too.** Rembrand Koning, Sharique Hasan and Aaron Chatterji studied start-ups' use of A/B testing tools and found that those adopting A/B testing saw performance improve 30% to 100% after a year. They attribute the gain to using testing as part of a broader learning strategy, which helped young firms scale good ideas and drop bad ones faster [18].

Lessons that don't transfer (yet)

- **Chasing tiny effects.** Enterprises profit from 0.5% lifts because they apply to billions of visits. A small team can't detect them and shouldn't try.
- **Many tests in parallel.** Running overlapping tests on low traffic slows every test. One lane, well chosen, beats three lanes starved of traffic.
- **Heavy custom platforms.** Microsoft, Spotify and Airbnb built their own. Small teams should borrow the methods through tools that already include them.

7. Can testing and optimization be automated?

Yes, partly, and more each year. There are three layers.

Layer 1: Automation already built into ad platforms

- **Google Ads Experiments.** Google's help center says the auto-apply feature for favorable results "is enabled by default" for some experiment types: if results favor the trial campaign, Google applies it and shifts 100% of traffic to it. You can turn it off at any time [12]. Note that the default confidence interval in experiment reports is 80% [12]. If a change is hard to undo, set a stricter level or review before it applies.
- **Google Conversion Lift.** Holdout studies that measure whether ads caused extra sales now use Bayesian methods with historical data. Google lists minimums of $5,000 in spend and 150 conversions from the treatment group and 65 from the control group for results to show [13].

- **Meta.** Meta's Experiments tool runs A/B tests of ads, and in June 2026 Meta announced AI-driven creative testing tools in Ads Manager, initially for select advertisers [24]. Meta's details change often, so check its Business Help Center for current behavior before relying on any automatic action.

Layer 2: Tests that shift traffic on their own ("bandits")

A **multi-armed bandit** is a test that moves traffic toward the better-performing version as results come in, instead of keeping a fixed 50/50 split. Scott describes bandits as experiments whose design "is modified as the experiment progresses to make the experiment as inexpensive as possible" [17].

- **Optimizely** offers two modes. Its Stats Accelerator moves traffic to reach a reliable answer faster; Optimizely says users "typically achieve statistical significance two to three times faster than standard A/B/n testing." Its bandit mode simply maximizes results and is described as "for optimization, not experimentation" [8].
- **GrowthBook**, an open-source tool, offers bandits and contextual bandits alongside its standard tests [14].

When to use which:

Use a standard test when...	Use a bandit when...
You need to *learn* why something works and apply it elsewhere	You need to *earn* during the test and the lesson matters less
The decision is lasting (offer, message, page)	The decision is short-lived (a promotion, a holiday ad)
You need an accurate estimate of the lift	You have several creative variations and want the best one fast

Bandits have a statistical catch. Optimizely's documentation explains that shifting traffic over time can bias results, even flipping a winner and a loser, unless the tool corrects for it, which Optimizely does with a weighted method [8]. Use bandits inside tools that handle this, not homemade rules.

Layer 3: Rules that act on thresholds

The next step is an automated decision engine: the written decision rule from Step 5, turned into code that watches every running test. A reasonable design:

Trigger	Automatic action	Person needed?
Split check fails	Pause the test and alert	Yes, to diagnose
A guardrail gets clearly worse (for example cost per lead up beyond its stopping bound)	Stop the variant and return traffic to the original	Review afterward
Variant passes the pre-set win bar on the main measure	Ramp it to 100% for easy-to-reverse changes (ads, creative)	Review for anything hard to undo
Futility bound reached	End the test, log "no meaningful difference," start the next test in the queue	No
Lift far larger than history suggests	Hold and alert (Twyman's law)	Yes
Maximum run time reached	End, log the estimate and range, re-score the backlog	No

This is where mid-size data teams can win back the tempo enterprises get from parallel testing: no idle days between tests, no forgotten tests running for months, and no "winner" applied without passing its own written rule.

What should stay human

Automation should not choose what to test, and it should not be the last word on surprising or hard-to-undo results. Someone has to judge whether a hypothesis is worth the lane, whether a win makes sense, whether it fits the brand, and whether the data behind it is sound. At ROI Factory, a domain Expert in the Loop™ checks everything we deliver. In testing, that check sits at three points: approving the queue, approving anything hard to undo, and investigating any result that looks too good.

---

8. A worked example

This example is illustrative. The company and its numbers are made up; the calculations use the formula in Appendix A.

A funded B2B start-up gets about 8,000 website visitors a month and turns 2.5% of them into leads (about 200 leads a month). It also spends on search ads, which bring about 3,000 clicks a month at a 6% lead rate.

Sizing the lane (website, 95% confidence, 80% power):

Test length	Visitors	Leads	Smallest detectable lift
2 weeks	~3,700	~90	~58%
4 weeks	~7,400	~185	~41%
8 weeks	~14,700	~370	~29%
12 weeks	~22,100	~550	~24%

To detect a 10% lift, this site would need about 122,000 visitors in the test: more than a year. A 30% lift needs about 7 weeks; a 50% lift under 3 weeks.

The backlog, sorted:

Idea	Bucket	Why
Fix the demo form that fails on some phones	Decide	Clear bug, easy to reverse
Change button color	Watch	Likely effect far below 29%
Interview five recent buyers about why they chose us	Research	Feeds better hypotheses
Lead with "live in two weeks" instead of feature list	Test (rung 3)	Big plausible effect, backed by sales-call notes
Free assessment instead of "book a demo"	Test (rung 2)	Offer change; big plausible effect
Three new ad headlines	Test in ads (rung 6)	Ads have the traffic; use Google's built-in experiments

Scoring the two website tests:

Idea	Reach (leads/mo)	Lift if it works	Chance it works	Lane-weeks	Score
Free assessment offer	200	50%	30%	4	7.5
New lead message	200	30%	40%	8	3.0

The offer test goes first: it is no more likely to work, but it would resolve in half the time. The message test can run in ads at the same time, since ads are a separate lane with their own traffic.

The rule for the offer test: main measure is leads; guardrail is the share of leads that become qualified; 90% confidence (the change is easy to reverse); weekly sequential checks; stop early for a clear win, clear harm or futility; maximum eight weeks.

9. The Mid-Size Data™ Playbook

1. **Fix measurement first.** Every lead and sale counted once, with its source; check the split.
2. **Size your lane.** Know the smallest lift you can detect in 2, 4 and 8 weeks.
3. **Sort every idea:** Decide, Research, Test or Watch.
4. **Score tests by value per lane-week:** (Reach × Lift × Chance) ÷ Lane-weeks.
5. **Climb the ladder:** offer and audience before pages; pages before details.
6. **Write the decision rule before launch:** one main measure, guardrails, confidence for the risk, stopping rules, actions for each outcome.
7. **Read fast, honestly:** sequential checks, a Bayesian expected-loss rule if your tool has it, variance reduction where you have history, earlier measures you've validated.
8. **Move creative tests to ad platforms** and use their built-in experiments.
9. **Automate inside the rules:** auto-stop for harm, auto-end for futility, auto-ramp easy-to-reverse wins, alert on surprises.
10. **Keep a person on the queue and on anything surprising or hard to undo.**

How ROI Factory helps

ROI Factory engineers go to market systems and automations for start-up founders and CMOs. For testing, that means we size your lane, build and score the backlog, write the decision rules, set up the tools and the automation, and run the tests one after another without idle weeks between them. A domain Expert in the Loop™ checks everything we deliver.

To talk about your testing program, visit roifactory.com.

Appendix A: The math behind the tables

For a test comparing two lead rates, the number of visitors needed in **each** group is approximately:

> $n = (z_1 + z_2)^2 \times 2 \times p \times (1 - p) \div \delta^2$

where p is the current lead rate, δ is the absolute difference you want to detect (current rate × relative lift), $z_1 = 1.96$ for 95% confidence (1.645 for 90%), and $z_2 = 0.84$ for 80% power. With 95% confidence and 80% power, $(z_1 + z_2)^2 \times 2 \approx 15.7$, which is where Evan Miller's "16 × variance ÷ effect²" rule of thumb comes from [5].

Rearranged, the smallest detectable relative lift for a test with n visitors per group is:

> $\text{lift} = \sqrt{[\ (z_1 + z_2)^2 \times 2 \times (1 - p) \div (p \times n)\ ]}$

Because $p \times n$ is roughly the number of leads in each group, the detectable lift depends mainly on the count of leads or sales, which is why that count drives Section 2. These formulas are standard approximations; dedicated calculators give slightly different numbers.

Appendix B: Plain-words glossary

Term	Plain meaning
A/B test	Show two versions to randomly split groups and compare results
A/A test	Show the same version to both groups to check the setup
Lead rate	The share of visitors who become leads
Lift	How much better (or worse) the new version did, as a percentage of the old
Minimum detectable effect	The smallest real lift a test of a given size will reliably catch
Power	How often a test catches a real lift of the size you planned for (80% is common)
Confidence level	How strict the test is about false wins (95% means about a 1-in-20 chance of a false win when there's no real difference)
Peeking	Checking results repeatedly and stopping on a good number, which inflates false wins
Sequential test	A test designed for checking on a schedule, with stricter bars at early looks
Futility stop	Ending a test early because a meaningful lift has become very unlikely
Bayesian	A way of stating results as probabilities ("92% chance B is better") that can include past knowledge
Expected loss	How much you'd expect to lose if you pick the wrong version
Variance reduction (CUPED)	Using each person's past behavior to remove noise from test results
Guardrail measure	A number that must not get worse, even if the main measure improves
Surrogate / earlier measure	An early step that predicts the outcome you care about
Multi-armed bandit	A test that automatically shifts traffic toward the better version
Sample ratio mismatch	The traffic split doesn't match the plan, a sign the test is broken

Sources

All links checked November 2026.

1. Kohavi, R. et al. (2012). *Online Controlled Experiments: Introduction, Learnings, and Humbling Statistics* (ACM RecSys keynote slides). https://ai.stanford.edu/~ronnyk/2012-09ACMRecSysNR.pdf
2. Kohavi, R., Deng, A., & Vermeer, L. (2022). "A/B Testing Intuition Busters: Common Misunderstandings in Online Controlled Experiments." *KDD 2022.* https://exp-platform.com/abtestingintuitionbusters/ (success-rate figures as presented in the authors' accompanying slides)
3. Blanding, M. (2019). "Creating the Experimentation Organization" (interview with Stefan Thomke). *HBS Working Knowledge.* https://www.library.hbs.edu/working-knowledge/creating-the-experimentation-organization
4. Georgiev, G. (2022). "What Can Be Learned From 1,001 A/B Tests?" *Analytics-Toolkit.com.* https://blog.analytics-toolkit.com/2022/what-can-be-learned-from-1001-a-b-tests/
5. Miller, E. (2010). "How Not To Run an A/B Test." https://www.evanmiller.org/how-not-to-run-an-ab-test.html
6. Schultzberg, M., & Ankargren, S. (2023). "Choosing a Sequential Testing Framework — Comparisons and Discussions." *Spotify Engineering*. https://engineering.atspotify.com/2023/3/choosing-sequential-testing-framework-comparisons-and-discussions
7. Johari, R., Pekelis, L., & Walsh, D. "Always Valid Inference: Continuous Monitoring of A/B Tests." *Operations Research* (2022). https://arxiv.org/abs/1512.04922
8. Optimizely. "Stats accelerator overview." https://docs.optimizely.com/experimentation-strategy/docs/stats-accelerator-overview
9. Deng, A., Xu, Y., Kohavi, R., & Walker, T. (2013). "Improving the Sensitivity of Online Controlled Experiments by Utilizing Pre-Experiment Data" (CUPED). *WSDM 2013.* https://exp-platform.com/cuped/
10. Athey, S., Chetty, R., Imbens, G., & Kang, H. "The Surrogate Index: Combining Short-Term Proxies to Estimate Long-Term Treatment Effects More Rapidly and Precisely." NBER Working Paper 26463 (2019, revised 2024). https://www.nber.org/papers/w26463
11. Kameleoon (2022). "Ronny Kohavi shares how to accelerate innovation by getting results you can trust." https://www.kameleoon.com/blog/ronny-kohavi-getting-results-you-trust
12. Google Ads Help. "Monitor your experiments." https://support.google.com/google-ads/answer/6318747
13. Google Ads Help. "About Bayesian methodology in Conversion Lift." https://support.google.com/google-ads/answer/16627074
14. GrowthBook Documentation. "GrowthBook's Statistics." https://docs.growthbook.io/statistics/overview
15. Laja, P. (2016, updated 2026). "PXL: A Better Way to Prioritize Your A/B Tests." *CXL*. https://cxl.com/blog/better-way-prioritize-ab-tests/
16. Stucchio, C. (2014). "Bayesian A/B testing decision rule." https://www.chrisstucchio.com/blog/2014/bayesian_ab_decision_rule.html, and Stucchio, C. (2015). *Bayesian A/B Testing at VWO* (whitepaper). https://www.chrisstucchio.com/pubs/VWO_SmartStats_technical_whitepaper.pdf
17. Scott, S. L. (2015). "Multi-armed bandit experiments in the online service economy." *Applied Stochastic Models in Business and Industry*, 31. https://research.google/pubs/multi-armed-bandit-experiments-in-the-online-service-economy/
18. Koning, R., Hasan, S., & Chatterji, A. (2022). "Experimentation and Start-up Performance: Evidence from A/B Testing." *Management Science*, 68(9), 6434–6453. https://doi.org/10.1287/mnsc.2021.4209 (working paper: https://www.nber.org/papers/w26278)
19. Black Swan Farming. "Cost of Delay Divided by Duration" (on Reinertsen's CD3 / WSJF). https://blackswanfarming.com/cost-of-delay-divided-by-duration/
20. McBride, S. (2018). "RICE: Simple prioritization for product managers." *Intercom.* https://www.intercom.com/blog/rice-simple-prioritization-for-product-managers/
21. Balfour, B. "Strategize, Test, Measure: The Bullseye Framework" (on Weinberg & Mares, *Traction*). https://brianbalfour.com/essays/traction-the-bullseye-framework
22. Growth Method. "PXL framework" (on the origins of PIE and PXL). https://growthmethod.com/pxl-framework/
23. Ellis, S. (2015). "Rhythm Is A Hacker" (WMD 2015 slides). https://www.slideshare.net/slideshow/wmd-qualaroo-sean-ellis/47632756
24. Hutchinson, A. (2026). "Meta launches advanced AI ad features." *Social Media Today*, June 23, 2026. https://www.socialmediatoday.com/news/meta-launches-advanced-ai-ad-features/823591/
25. Kohavi, R. (2016). *Online Controlled Experiments* (talk slides, "Hierarchy of Evidence and Observational Studies"). https://www.exp-platform.com/Documents/2016-12OnlineControlledExperimentsYahooTechPulseNR.pdf

26. Kohavi, R., Tang, D., & Xu, Y. (2020). *Trustworthy Online Controlled Experiments: A Practical Guide to A/B Testing*. Cambridge University Press. Chapter: "Metrics for Experimentation and the Overall Evaluation Criterion." https://www.cambridge.org/core/books/trustworthy-online-controlled-experiments/metrics-for-experimentation-and-the-overall-evaluation-criterion/4EA73D169EC43B58991D6824717E8FD3

© 2026 ROI Factory.