LAD

A/B Test Checklist

Everything to decide, build, verify, run and analyse so that an A/B test produces a result you can actually trust. Twelve stages, from writing the hypothesis to removing the code after rollout.

88 checks **12** stages **28** marked as highest leverage datalad.co.uk

How to use it. Stages 01 to 08 are pre-launch and are where almost every invalid test is created; do not skip them because the build looks finished. Stages 09 to 12 are the discipline that keeps a live test valid and turns the number into a decision. Items in **bold** with a heavier box are the ones that most often invalidate a test outright, so if you check nothing else, check those. The checklist is tool-agnostic and works for client-side, server-side and feature-flag testing alike.

01 Hypothesis and business case

Before anything is built

- [ ] **The hypothesis is written down in one sentence**
  Because [evidence], changing [X] will [effect], measured by [metric]
- [ ] **It cites evidence, not a hunch**
  Analytics, session recordings, support tickets, user research
- [ ] **It is falsifiable**
  You can state in advance what result would disprove it
- [ ] **The change is big enough to be worth measuring**
  A button shade rarely moves a metric past the noise
- [ ] **The expected value justifies the effort**
  Traffic times baseline times a plausible lift, against build cost
- [ ] **The question has not already been answered**
  Check the test repository before you build anything

02 Metrics

Decide what winning means, in advance

- [ ] **One primary metric, chosen before launch**
  Two primary metrics means no decision rule at all
- [ ] **The primary metric sits close to the change**
  A checkout test measures checkout, not annual retention
- [ ] **Secondary metrics listed and their role stated**
  Context and explanation, never the basis of the decision
- [ ] **Guardrail metrics defined**
  Revenue, refunds, page speed, support volume, unsubscribes
- [ ] **Every metric has a written definition**
  What exactly counts as a conversion, and over what window
- [ ] **Unit of analysis matches unit of randomisation**
  Randomise by user, then analyse by user, not by session

03 Statistical design

The maths that decides how long you wait

- [ ] **Baseline rate measured from recent, comparable data**
  Same audience, same season, same traffic mix

- [ ] **MDE set by what would change the decision**
  Not by what the sample size calculator makes achievable
- [ ] **Sample size calculated before launch**
  Power 80%, alpha 5% unless you have a reason to differ
- [ ] **Duration covers at least one or two full business cycles**
  Weekday and weekend behaviour differ, so never stop mid-week
- [ ] **Multiple comparisons accounted for**
  More variants or more metrics means a higher false positive rate
- [ ] **A sequential or Bayesian method if you intend to look early**
  Peeking at a fixed-horizon test invalidates the p-value
- [ ] **One-tailed or two-tailed decided and justified**
  Two-tailed by default: a change can hurt as well as help
- [ ] **The stopping rule is written down**
  Sample reached, or duration reached, whichever comes second

04 Test design

What varies, and for whom

- [ ] **Only the intended element differs between variants**
  Any second change makes the result uninterpretable
- [ ] **The variant is a genuine test of the hypothesis**
  If it does not test the stated mechanism, rewrite one of them
- [ ] **Eligibility rules are explicit**
  Pages, devices, geographies, new versus returning, logged-in state
- [ ] **Exclusions defined**
  Internal traffic, employees, bots, staging and QA sessions
- [ ] **Traffic allocation set deliberately**
  50/50 gives the most power; uneven splits need more traffic
- [ ] **Randomisation is by user and persists**
  The same person must see the same variant on every visit
- [ ] **Conflicts with other running tests checked**
  Overlapping surfaces produce interaction effects nobody can untangle
- [ ] **A holdout considered for long-term effects**
  Especially for pricing, messaging and loyalty changes

05 Implementation

Building it without breaking the page

- [ ] **Server-side or feature-flagged where possible**
  Fewer flicker problems, and it survives ad blockers
- [ ] **No flash of original content**
  Anti-flicker handling, or the test measures the flicker
- [ ] **The variant introduces no layout shift**
  A CLS change is itself a confound
- [ ] **Single-page app route changes re-evaluate correctly**
  A test that only fires on hard loads misses most of the session
- [ ] **Performance impact measured, both variants**
  Compare LCP and INP; a slower variant loses for the wrong reason
- [ ] **No new console errors in either variant**
  Check on real devices, not only in the editor preview
- [ ] **Behaviour under denied consent is defined**
  Either excluded from the test, or tracked in a compliant way

- [ ] **The code is reviewed by someone else**
  The author cannot see their own assumption
- [ ] **Rollback is a single toggle**
  And someone other than the author knows where it is

06 QA before launch

Both variants, everywhere

- [ ] **A way to force a variant for testing**
  A query parameter or a cookie override, documented for the team
- [ ] **Both variants checked on real iOS and Android devices**
  Emulators miss Safari behaviour and touch targets
- [ ] **Chrome, Safari, Firefox and Edge**
  Safari finds what the others do not
- [ ] **Logged-in and logged-out states both checked**
  Personalised content is where variants usually break
- [ ] **Edge cases exercised deliberately**
  Empty cart, long names, out of stock, error and loading states
- [ ] **Accessibility is unchanged or better**
  Contrast, focus order, labels and reduced motion, in both variants
- [ ] **Localised versions checked if the page is translated**
  Text expansion breaks layouts that were fine in English
- [ ] **Screenshots of control and variant saved**
  You will need them at analysis, and the page will have moved on

07 Tracking

If the measurement is wrong, nothing else matters

- [ ] **An exposure event fires once per user, per variant**
  Only on users who actually saw the tested surface
- [ ] **Exposure fires at the point of exposure, not page load**
  Otherwise the sample includes people who never reached it
- [ ] **The conversion event verified end to end**
  A real conversion, watched in DebugView or the equivalent
- [ ] **Variant assignment sent to analytics as a dimension**
  So the test can be analysed outside the testing tool
- [ ] **Revenue tracked with the right value and currency**
  Net of tax and shipping, decided in advance and applied to both
- [ ] **Segment dimensions captured**
  Device, source, new versus returning, and any audience that matters
- [ ] **Consent mode and server-side parity checked**
  Know which users are missing from the data, and why
- [ ] **A test conversion completed before launch**
  In both variants, with the result confirmed in the report

08 Data quality

The checks that catch a broken test

- [ ] **A sample ratio mismatch check is planned**
  With a threshold, and an owner who runs it daily
- [ ] **Bot and internal traffic filtered**
  Otherwise both arms are diluted, and unevenly
- [ ] **Testing tool and analytics agree within a tolerance**
  Reconcile the counts before launch, not during analysis

- [ ] **Cross-device and cross-browser dilution understood**
  Accept the limit, or use a logged-in-only audience
- [ ] **Redirect tests checked for speed and referrer effects**
  A redirect variant is slower by construction
- [ ] **Data retention and access confirmed**
  You can still pull the raw data when you analyse it

09 Launch

Getting it live without surprises

- [ ] **Stakeholders told what is running, where, and for how long**
  Marketing, support, sales and anyone who reads the dashboards
- [ ] **Ramped: a small percentage first, verified, then full**
  Ten minutes at 5% catches most implementation faults
- [ ] **Start timing avoids campaigns, launches and holidays**
  Anything unusual in the period becomes a confound
- [ ] **The end date is in a calendar, with an owner**
  Tests that nobody owns run for months and decide nothing
- [ ] **Emergency stop criteria written down**
  Which metric, what threshold, who decides, and how fast
- [ ] **Other changes to the tested surface are frozen**
  A mid-test redesign ends the test, whether you notice or not

10 While it runs

Discipline, mostly

- [ ] **First-hour smoke check**
  Exposures in both arms, conversions arriving, no error spike
- [ ] **Sample ratio mismatch checked daily**
  An SRM means the assignment is broken; stop and fix it
- [ ] **Guardrail metrics watched daily**
  Stop early for harm, never stop early for a win
- [ ] **No decisions on partial data**
  Unless you designed for sequential analysis, results before the end are noise
- [ ] **Novelty and primacy effects considered**
  Returning users react to change itself for the first week or two
- [ ] **Anything that could confound is logged**
  Outages, campaigns, press, competitor moves, seasonality
- [ ] **Neither variant is edited mid-test**
  A change resets the experiment, so restart it instead

11 Analysis

Reading the result honestly

- [ ] **Run to the planned sample and duration**
  Not until it happens to be significant
- [ ] **SRM checked before believing anything else**
  A failed SRM invalidates the whole result, however pretty
- [ ] **Primary metric first, then guardrails, then secondaries**
  In that order, decided before you looked
- [ ] **Confidence intervals reported, not only p-values**
  The range of plausible effects is the useful part
- [ ] **Practical significance judged as well as statistical**
  A real 0.2% lift may still not be worth shipping

- [ ] **Segment findings treated as hypotheses**
  Slice enough segments and one will always look significant
- [ ] **Outliers inspected and their influence stated**
  One enterprise order can carry an entire revenue result
- [ ] **A flat result is recorded as a result**
  Knowing a change does nothing is worth the traffic it cost

12 Decision and documentation

Turning a number into a change

- [ ] **The decision recorded: ship, kill or iterate**
  With the reasoning, not only the outcome
- [ ] **Rollout plan agreed, including the losing variant's removal**
  Half-rolled-out tests are a lasting source of confusion
- [ ] **Test code and flags removed after rollout**
  Dead experiment code accumulates and slows every page
- [ ] **The result logged in the test repository**
  Hypothesis, design, data, decision, and the date
- [ ] **Screenshots and the analysis attached**
  So the result can be re-read in a year and still make sense
- [ ] **The learning shared, including the losses**
  Failed tests are the ones that stop the team repeating them
- [ ] **A follow-up hypothesis written while it is fresh**
  The best next test is usually obvious on the day
- [ ] **Effect re-measured after full rollout**
  Confirm the lift survives outside the experiment