
A RESOURCE FOR DESIGN SYSTEM AND DESIGNOPS TEAMS

AARRR funnel for design systems.

Measure adoption the way growth teams measure products.

Define the events. Count the stages. Fix the weakest one.

25

checks, each with its evidence

05

stages, awareness to revenue

11

metrics with SQL that runs

Adoption is a funnel. Measure it like one.

In 2007 Dave McClure split a product's growth into five conversion stages: acquisition, activation, retention, referral and revenue [S01]. The essay behind this guide applies them to design system adoption, with one change: awareness replaces acquisition, because everyone who could use an internal system already works for you.

Few teams measure adoption as a sequence. In zeroheight's 2026 report, 41% of 147 practitioners measure adoption and 38% measure component usage in code [S06]. Coverage alone says adoption stalled, not where. A funnel shows which stage leaks.

The kit: a metric file with 11 definitions (owner, source, target and a SQLite query each), a zero-dependency calculator for CSV event exports, and a seeded sample of 60 engineers in five teams over 13 weeks. Every number in this guide is kit output on that sample.

Version 1.0 / Sources checked 24 September 2026

Field Guide 16 of the Design × AI series. Pairs with Field Guides 17 (time-to-first-component benchmark) and 18 (component usage Slack bot); all three use the same organisation, Acme, and its `@acme/ui` library. Verified 24 September 2026 with Node 22.22 and the SQLite 3.51.0 CLI.

Practical guidance, not a standard. The sample data is synthetic: it demonstrates the method and proves nothing about real teams. Targets marked essay come from the essay Growth loops for design system adoption; kit defaults are placeholders for your own baseline. Prepared with AI assistance and edited by hand.

START HERE

Five stages, one event export.

Each stage asks one question and is answered by events you can observe.

Stage	McClure [S01]	Design system meaning	Qualifying events	Primary metric
Awareness (was acquisition)	Users come to the site	Engineers know the system exists	<code>docs_visit</code> , <code>channel_join</code> , <code>office_hours_attend</code> , <code>package_install</code>	AWR-1, share of eligible
Activation	A happy first visit	A first merged PR that imports <code>@acme/ui</code>	first <code>ds_pr_merged</code>	ACT-1, share of aware
Retention	Users come back	Another system PR 14 to 28 days later	<code>ds_pr_merged</code>	RET-1, share of measurable
Referral	Users refer others	A peer recommendation, answer or contribution	<code>referral_comment</code> , <code>channel_answer</code> , <code>contribution_merged</code>	REF-1, share of activated
Revenue	Users pay	Team outcomes: coverage and defects	<code>coverage_snapshot</code> , <code>ui_defect</code>	REV-1, teams at 70% coverage

Starting from nothing

Section 01: a roster and an event taxonomy. Export three months before setting targets.

You already track coverage

Coverage is revenue, the last stage to move. Add activation, the leading signal.

Leadership wants one number

Section 04: team outcomes with a stated formula. No story points.

Adoption has stalled

Find the weakest stage. Field Guide 17 diagnoses activation; 18 feeds referral back into awareness.

Label	Meaning
DATA	A field in a kit data file: metric definitions, roster or event export.
SCRIPT	Computed or enforced by <code>funnel.mjs</code> or <code>sql-report.mjs</code> .
PROTOCOL	How the measurement is run, so numbers stay comparable.
PRIVACY	Protects the people measured. Adoption data describes teams, not performance.
LOOP	Feeds a later stage back into an earlier one.
PRACTICE	A working method with a review signal, not a hard gate.

SECTION 01

Define who counts, and what happened.

A funnel is a denominator and a list of events. Get either wrong and every rate after it is wrong in a way no chart shows.

Suggested owners: DesignOps lead + design system product manager

☐ The population comes from a roster, not from usage

F01 **DATA** **SCRIPT**

Count against everyone who could use the system: engineers who build UI, from an HR or GitHub team export. Leave the system team out; they would inflate every stage.

Evidence: `roster.csv` with `builds_ui`. The calculator counts 60 of the sample's 64 engineers as eligible.

Kit: population POP.

☐ One event taxonomy, each event defined and sourced

F02 **DATA** **SCRIPT**

Eleven events cover the five stages, each with a definition and its system of record. A metric that references an undefined event does not validate.

Evidence: `events[]` in `aarr-metrics.json`; `validateMetrics()` fails on unknown events.

Kit.

☐ Events are pseudonymous and carry a team

F03 **PRIVACY** **DATA**

The export holds `e017`, never a name; managers keep the key. Reports show stages and teams, never a ranked list of people. No single activity count captures productivity.

Evidence: Columns `timestamp`, `actor`, `team`, `event`, `value`; no names in the report repo.

S09.

☐ Every metric names an owner, a cadence and a source

F04 **DATA** **SCRIPT**

A number nobody owns is a number nobody acts on. The owner is a role; the source is the system of record.

Evidence: `validateMetrics()` fails on a missing `owner`, `cadence` or `dataSource`.

Kit.

☐ Every target says where it comes from

F05 **DATA** **SCRIPT**

From the essay, your baseline or a kit default. A metric with no target says why: trend only, or baseline first.

Evidence: `target.basis` required; a `null` target needs `targetNote`. 6 of 11 metrics have targets, 5 from the essay.

Kit.

The denominator is a decision. Write it down before you chart anything.

SECTION 02

Known, then used.

Awareness is cheap to measure and easy to overstate. Activation is what the essay calls the north star: the system doing real work for someone.

Suggested owners: DesignOps lead + design system engineering lead

☐ Awareness counts observable signals, calibrated by a survey

F06 **DATA** **PROTOCOL**

A docs visit, a channel join, office hours or a package install makes an engineer aware. Twice a year, ask whether people know the system and compare; Pinterest's Gestalt team pairs tooling data with surveys the same way [S04].

Evidence: AWR-1 on the sample: 53 of 60, 88.3%, below the essay's 90%.

Essay target. S04.

☐ Activation is a merged pull request, not an install

F07 **DATA**

An install is intent. A merged pull request whose diff adds an `@acme/ui` import is a result.

Evidence: ACT-1 on the sample: 47 of 53 aware engineers, 88.7%, against the essay's 80%.

Essay. Kit output.

☐ Stages are strict, and skips are tracking gaps

F08 **SCRIPT**

An engineer counts for a stage only after passing the one before. A system pull request with no awareness event means a signal is missing from the export: report it, fix the tracking.

Evidence: The calculator reports 2 tracking gaps on the sample and leaves them out of every rate.

Kit: `funnel.mjs`.

☐ New hires get their own activation clock

F09 **DATA** **SCRIPT**

The essay's target: 80% activated within 7 days of onboarding. Hires who started under 7 days before the window ends are not measurable yet, so they are left out, not failed.

Evidence: ACT-2: 4 of 5 measurable hires. The sixth started 26 August and is excluded.

Essay target. Kit output.

☐ Field time and benchmark time stay separate

F10 **PROTOCOL**

Calendar days from onboarding to first merged PR include meetings and weekends; task minutes in a session do not. Report both, never add them.

Evidence: ACT-3: a median of 5.24 calendar days over 5 hires. Task minutes: Field Guide 17.

Kit output. Field Guide 17.

Awareness says they found the door. Activation says they walked through it.

SECTION 03

Back again, and bringing others.

Retention separates a system people tried from one they build with. Referral is where the funnel starts to feed itself.

Suggested owners: Design system product manager + community lead

☐ The retention window fits bursty UI work

F11 **DATA**

A week on backend work is not churn. RET-1 looks for another system pull request between day 14 and day 28 after activation.

Evidence: `window: { fromDay: 14, toDay: 28 }` on RET-1; change it in the file, not the script.

Kit default window. Essay target: 80% at week 4.

☐ Recent activations are left out, not counted as lost

F12 **SCRIPT**

Someone activated 10 days ago cannot be retained at week 4 yet. Counting them as lost punishes every week you onboard well. Report weekly active engineers beside the population, not alone.

Evidence: RET-1: 31 of 38 measurable, 81.6%; 9 too recent. RET-2: 22 active of 60 eligible.

Kit output.

☐ Referral is an act you can see

F13 **DATA**

A review comment pointing to the right component, a peer answer in the channel, a merged contribution. The essay sets no referral target; set one after a quarter.

Evidence: REF-1: 13 of 47 activated engineers, 27.7%, reported as a trend.

Essay. Kit output.

☐ Outside contributions are counted and thanked

F14 **LOOP**

A merged pull request to the system from a product engineer is the strongest referral there is. Thank it in public as a team milestone.

Evidence: REF-2: 4 merged contributions. Field Guide 18's bot posts them.

Kit output. Field Guide 18.

☐ Champions are named only with consent

F15 **PRIVACY** **LOOP**

The essay recommends recognising engaged engineers with early access. Do it with their agreement. A leaderboard of individuals turns recognition into surveillance.

Evidence: A consent note beside each champion in the report; no per-person rankings.

Essay. S09.

Retention proves the first use was not an accident. Referral makes the next one cheaper.

SECTION 04

Revenue means outcomes, at team level.

Nobody buys seats of an internal system. The organisation pays for consistency, quality and speed, and those belong to teams.

Suggested owners: Design system lead + engineering director

☐ Revenue is measured per team, never per engineer

F16 **PRIVACY** **DATA**

Coverage and defects belong to a product surface. Attributing them to people measures who works on legacy code.

Evidence: `coverage_snapshot` and `ui_defect` events carry a team as the actor.

Kit. S09.

☐ Coverage has a written numerator and denominator

F17 **DATA**

Pinterest divides Gestalt layers by all layers in recent handoff files [S04]; Mews divides system DOM elements by all elements in production [S05]. Both are valid; mixing them is not. The kit uses code instances, `ds/total`, from Field Guide 18's scanner or react-scanner [S07].

Evidence: REV-1: 3 of 5 teams at 70% or more. Accounts (61.2%) and Billing (54.9%) are below.

S04, S05, S07. Essay target of 70%.

☐ Quality is compared with footprint

F18 **SCRIPT**

Counting use misses whether it is correct [S03]. Divide the system's share of UI defects by its share of instances; below 1.0, system components cause fewer defects than their footprint.

Evidence: REV-2: 0.48. The system is 72.7% of instances and 35% of 40 defects.

S03. Kit default target.

☐ No story points

F19 **PRACTICE** **SCRIPT**

Story points are estimates relative to one team's history. They do not compare across teams or quarters, so a before-and-after on points proves nothing about the system.

Evidence: `validateMetrics()` rejects a metric defined in story points.

Recommended practice. S09.

☐ Hours saved shows its formula or stays out

F20 **PRACTICE**

Adoption is a proxy [S08], and hours saved is a model on top of it. If you report one, print its inputs and formula beside it and call it an estimate.

Evidence: Every time-saved figure lists its inputs.

S08.

Report outcomes you can defend in the room, with the formula on the slide.

SECTION 05

Run it weekly. Act on one stage.

The essay says track weekly and share monthly. The rest is discipline: the same definitions every week, and one decision per report.

Suggested owners: Design system product manager

☐ **The metric file is versioned like code**

F21 **PROTOCOL** **DATA**

Changing an event, a window or a target changes the numbers. Bump `version`, note it in the report and mark the break in any chart.

Evidence: `version` in `aarr-metrics.json`, reviewed in version control.
Kit.

☐ **The calculator and the SQL agree**

F22 **SCRIPT**

Node or warehouse, the numbers must match. `sql-report.mjs` writes a SQLite script from the same metric file, so the queries have one source of truth.

Evidence: On the sample, all 11 metrics match between `funnel.mjs` and `sqlite3` 3.51.0.
Kit. S10 for `median()`.

☐ **Small bases show counts, not only rates**

F23 **SCRIPT**

On a team of 9, one person moves a rate by 11 points. The calculator flags rates on fewer than 10 people.

Evidence: `--team Accounts` flags RET-1: 7 of 9, 77.8% (small base).
Kit output.

☐ **Each report ends with one decision**

F24 **PRACTICE**

Name the weakest stage, one change, an owner and a date. A report that ends with a chart has informed nobody.

Evidence: The Decisions section of `templates/weekly-funnel.md` is filled. On the sample: awareness, and the two teams below coverage target.
Kit template.

☐ **Later stages feed the earlier ones**

F25 **LOOP**

Growth loops reinvest a stage's output as the next cycle's input [S02]. Milestones posted in the channel (Field Guide 18) bring new people in; a revenue story leadership believes funds the next onboarding work.

Evidence: Channel joins in the week after a celebration post, visible in the export.
S02. The essay's loop: visible metrics, buy-in, resources, discovery.

A funnel shows where adoption leaks. A loop is what refills it.

APPENDIX A

One file defines every number.

`metrics/aarr-metrics.json` holds the population, the event taxonomy, staging SQL and 11 metrics. The calculator reads its events, windows and targets; `sql-report.mjs` reads its queries.

metrics/aarr-metrics.json (excerpt: RET-1)

```
{
  "id": "RET-1",
  "primary": true,
  "name": "Week-4 retention",
  "definition": "Activated engineers with another merged design system pull request between day 14 and day 28 after activation, over activated engineers whose activation is at least 28 days before the end of the window.",
  "events": ["ds_pr_merged"],
  "window": { "fromDay": 14, "toDay": 28 },
  "query": "SELECT COUNT(*) AS measurable, SUM(EXISTS (SELECT 1 FROM win w WHERE ...)) AS retained FROM activated a WHERE julianday(a.first_at) + 28 <= julianday(:end);",
  "dataSource": "GitHub pull request diffs",
  "owner": "Design system product manager",
  "cadence": "weekly",
  "target": { "value": 0.8, "unit": "share", "direction": "at least", "basis": "essay" }
}
```

The query is shortened here; the file has it in full.

Metric	Owner	Target	Basis
AWR-1 Awareness rate	DesignOps lead	90% of eligible	essay
ACT-1 Activation rate	DS engineering lead	80% of aware	essay
ACT-2 New-hire activation in 7 days	DS engineering lead + onboarding	80%	essay
RET-1 Week-4 retention	DS product manager	80% of measurable	essay
REV-1 Teams at target coverage	DS lead	70% per team	essay
REV-2 Defect ratio	DS lead + QA lead	below 1.0	kit default
AWR-2, ACT-3, RET-2, REF-1, REF-2	various	none: trend or baseline first	targetNote

APPENDIX B

The funnel on the sample.

Sixty eligible engineers, five teams, 471 events from 1 June to 30 August 2026. Synthetic, seeded and reproducible with `node scripts/generate-sample.mjs`.

terminal

```
$ node scripts/funnel.mjs data/events.sample.csv --roster data/roster.sample.csv \
  --from 2026-06-01 --to 2026-08-30
AARRR funnel for @acme/ui, 2026-06-01 to 2026-08-30
Eligible engineers: 60 across 5 teams

Stage                Count   Base   Rate   Target  Status
AWR-1 Awareness rate    53     60   88.3%   90.0%  below target
ACT-1 Activation rate    47     53   88.7%   80.0%   met
RET-1 Week-4 retention   31     38   81.6%   80.0%   met
REF-1 Referral rate     13     47   27.7%    none  trend
REV-1 Teams at target coverage  3      5   60.0%  70% each 2 teams below
  9 activated engineers are too recent for week-4 retention
  and are left out of its base.

ACT-2 New-hire activation within 7 days 80.0% 4 of 5 measurable hires (met)
ACT-3 Field time to first component    5.24 median calendar days, 5 activated hires
REV-2 Defect ratio                     0.48 40 defects; system share 0.35 of
                                         defects, 0.727 of instances (met)

Tracking gaps: 2 eligible engineers merged design system PRs with no
awareness event. Fix the tracking; they are not counted.
```

Excerpt, wrapped to fit the page. `--markdown` writes the weekly report; `--json` feeds a dashboard; `--team` filters to one team.

terminal

```
$ node scripts/sql-report.mjs --from 2026-06-01 --to 2026-08-30 | sqlite3 :memory:
--- AWR-1 Awareness rate
  aware = 53
eligible = 60
  rate = 0.883
--- RET-1 Week-4 retention
measurable = 38
  retained = 31
--- REV-2 Defect ratio
  defects = 40
ds_defect_share = 0.35
ds_usage_share = 0.727
  ratio = 0.48
```

Same metric file, same export, same numbers (item F22). Excerpt of 3 of the 11 metrics.

KEEP WITH THE WEEKLY REPORT

Leave a measurement record.

One record per reporting period, so next quarter's reader knows exactly what was counted and what changed.

Reporting window (from, to)	Metric file version
Eligible population (count, roster date)	Event export (source systems, row count)
Tracking gaps found and fixed	Weakest stage this period
Decision (change, owner, date)	Coverage definition in use
Champions thanked (with consent)	Next review

Change the definition, not the history.

When a definition changes, bump the metric file version and restart the trend line. Do not recompute old periods silently.

SOURCES / MAINTENANCE

Keep the guide current.

Sources checked 24 September 2026. Survey figures are as published by the survey owner. Nathan Curtis's EightShapes articles on adoption could not be fetched from Medium on the check date; his view is cited through the Omlet round-up that quotes him.

S01 / Dave McClure, Startup Metrics for Pirates

The AARRR stages and per-stage conversion (2007 slides).

<https://www.slideshare.net/slideshow/startup-metrics-for-pirates-long-version/89026>

S02 / Reforge, Growth Loops are the New Funnels

Balfour, Winters, Kwok and Chen: loops reinvest output as input (31 July 2018).

<https://www.reforge.com/blog/growth-loops>

S03 / Omlet, How design system leaders define and measure adoption

Curtis, Onori, Kavcic and others on adoption stages and correct use (May 2024).

<https://omlet.dev/blog/how-leaders-measure-design-system-adoption/>

S04 / Figma blog, How Pinterest's design systems team measures adoption

Gestalt layers over total layers in recent handoff files; twice-yearly surveys (Feb 2023).

<https://www.figma.com/blog/how-pinterests-design-systems-team-measures-adoption/>

S05 / Mews, Building a design system adoption metric from production data

System DOM elements over all elements, measured in production (June 2025).

<https://developers.mews.com/design-system-adoption-metric-building/>

S06 / zeroheight, Design Systems Report 2026

147 practitioners; 41% measure adoption, 38% component usage in code.

<https://report.zeroheight.com/>

S07 / react-scanner

Static component usage counts for React; version 1.2.0. One way to produce coverage snapshots.

<https://github.com/moroshko/react-scanner>

S08 / Knapsack, Why design system adoption isn't the true measure of success

Adoption as a proxy metric; outcomes over usage (December 2024).

<https://www.knapsack.cloud/blog/why-design-system-adoption-isnt-the-true-measure-of-success>

S09 / Forsgren et al., The SPACE of Developer Productivity

ACM Queue, 2021: productivity is not captured by one metric or by activity alone.

<https://www.microsoft.com/en-us/research/publication/the-space-of-developer-productivity-theres-more-to-it-than-you-think/>

S10 / SQLite, the percentile extension

median() and percentile() with linear interpolation; in the amalgamation since 3.51.0.

<https://sqlite.org/percentile.html>

Maintenance: review targets each quarter against your own baseline and replace kit defaults once you have one. Recheck the zeroheight report when its next edition is published. Update the PDF, HTML, Markdown and JSON together.