A RESOURCE FOR DESIGN SYSTEM TEAMS

Deprecation policy with dates.

Announce with the replacement. Remove on the date.

Six months minimum. Majors only. CI keeps the promise.

22	06	10
items, each with its evidence	months of notice, at least	rules the CI checker enforces

A deprecation without a date is a wish.

The essay behind this guide sets the policy in five lines: announce with the replacement ready, give at least six months of notice, publish the migration guide on announcement day, track adoption, and remove only when usage is low. "This is not kindness," it says; it is how you avoid emergency rollbacks when teams miss announcements.

Mature platforms agree on the shape and differ on the numbers. SemVer requires a minor release for a deprecation and a major for the removal [S01]. Node.js moves APIs from documentation-only to runtime warnings to end-of-life [S02]. Kubernetes gives beta APIs nine months or three releases [S03]. Polaris asked for deprecations a month before a major [S09]. What they share is that the dates are written down before anyone needs them.

The kit writes them into a registry and lets CI hold you to them. The checker passes the Acme UI registry, reports 14 errors on a registry that breaks every rule, and starts failing the day after a removal date passes. DEP-0007, the `Button` `href` deprecation from RFC-0042 (Field Guide 07), is announced in the October 2026 train and removed in the April 2027 major window (Field Guide 08).

Version 1.0 / Sources checked 24 September 2026

Field Guide 09 of the Design × AI series, with 07 and 08 from the essay Enterprise UX that ships. Verified 24 September 2026: Node 22.22; jscodeshift 17.4.0 for the codemod (the only dependency); typescript-eslint 8.70 current for no-deprecated.

Practical guidance, not a standard. Six months and 5% are the essay's numbers; set your own in the registry policy. Acme UI and its usage counts are a worked example. Prepared with AI assistance and edited by hand.

START HERE

From announcement to removal.

Every deprecation moves through the same stages. The registry records which stage each one is in, and the dates that move it to the next.

Stage	What consumers see	Node.js	Atlassian
Intent (optional)	An RFC in discussion (Field Guide 07)		Intent to deprecate
`deprecated`	Docs, `@deprecated` JSDoc, lint, a dev-only console warning, a changelog line	Documentation-only, then runtime	Deprecated
`removed`	Gone in the major named on day one, on the date named on day one	End-of-Life	Removed after the period
`revoked`	The deprecation is withdrawn; the id stays	Reversed; the code is kept	

Node.js and Atlassian stages from S02 and S08. Owners: the design system lead owns the policy and Section 03; the component owner owns each entry, its notice and its codemod.

Deprecating something

Section 01, then `deprecation-notice.mjs DEP-NNNN` for every text you need. The replacement must already ship.

Setting the policy

Section 03. Put the minimum notice, the major windows and the usage threshold in the registry's `policy` block.

Adding the CI gate

Run `check-deprecations.mjs` on every pull request and on a daily schedule, so an overdue date fails even when nobody touches the code.

Helping teams migrate

Section 04. Measure usage, ship a codemod for every mechanical change, say why when there is none.

Label	Meaning
`REGISTRY`	A field in `deprecations.json`, validated by `deprecations.schema.json`.
`CHECK`	Enforced by `check-deprecations.mjs`; the evidence line names the rule (E1 to E10).
`SEMVER`	Traces to Semantic Versioning 2.0.0.
`SIGNAL`	Something a consumer sees: editor, lint, console, changelog, npm.
`CODEMOD` / `PRACTICE`	Automated migration, or a working method with a review signal.

SECTION 01

Announce with everything ready.

An announcement starts a clock. Start it when teams can act that day.

- [ ] **The replacement ships before the announcement**

  D01 REGISTRY CHECK

  `LinkButton` ships in the 4.8.0 release that deprecates `Button` `href`.

  Evidence: `replacement` names a shipped API, or `replacementReason` says why nothing replaces it (rule E7).

  Essay: announce with the replacement ready.

- [ ] **Announce in a minor or major, never a patch**

  D02 SEMVER CHECK

  Deprecating public API is a minor change. A deprecating patch surprises everyone who pins to patches.

  Evidence: Rule E3 on `announcedIn`; Field Guide 08's gate applies the same rule to changesets (C5).

  SemVer 2.0.0 rule 7 and its deprecation FAQ. S01.

- [ ] **Every deprecation gets a permanent id**

  D03 REGISTRY CHECK

  `DEP-0007` in the registry, changelog, JSDoc, console and RFC. Never reused or renumbered, even when revoked.

  Evidence: Schema pattern `DEP-NNNN`; rule E2 on duplicates.

  Node.js keeps a deprecation's identifier even when the deprecation is reversed. S02.

- [ ] **The migration guide exists on announcement day**

  D04 REGISTRY CHECK

  Why, what replaces it, the automated path and the manual one. Linked from every notice.

  Evidence: `migrationGuide` is required (rule E7).

  Essay; Polaris required the reason, alternatives, automated and manual migration steps. S09.

- [ ] **Every date is set on day one**

  D05 REGISTRY

  `announced`, `announcedIn`, `removalVersion`, `removalDate`: all required. "A future major" is not a date.

  Evidence: Schema validation (rule E1).

  Atlassian: no GA feature is removed without a clearly announced deprecation period. S08.

- [ ] **One entry generates every notice**

  D06 REGISTRY

  Notice, JSDoc line, console warning and changeset come from one entry, so they cannot disagree.

  Evidence: The generated notice equals `examples/DEP-0007.notice.md`; its changeset passes Field Guide 08's gate.

  Kit script.

Start the clock only when a team could finish the migration today.

SECTION 02

Make it impossible to miss.

Teams miss announcements. They do not miss a struck-through name, a lint error or a console warning.

- [ ] **`@deprecated` JSDoc on the declaration**

  D07 SIGNAL

  With the version, the replacement, the removal version and date, and the id. TypeScript surfaces the tag in completions and editors strike the name through.

  Evidence: Generated: `@deprecated Since @acme/ui 4.8.0. Use LinkButton. Removed in 5.0.0 on 13 April 2027 (DEP-0007).`

  TypeScript 4.0 `@deprecated` support. S05.

- [ ] **Lint turns the tag into an error in consuming repos**

  D08 SIGNAL

  `@typescript-eslint/no-deprecated` reports every use of a `@deprecated` declaration. It needs type information, so run it in the typed lint job.

  Evidence: Field Guide 03's kit runs the rule against its deprecated `color.link` token; this kit does not run ESLint. S06.

- [ ] **Development builds warn once; production stays silent**

  D09 SIGNAL

  Once per id per page load, in development only. A warning that repeats on every render gets filtered out; a warning in production changes behavior for end users.

  Evidence: `warnDeprecated()` in `src/warn-deprecated.mjs`: the kit's tests assert one call in development and none in production.

  React ships development-only warnings ahead of breaking changes; Primer shows deprecated components a warning. S07, S12.

- [ ] **Each train's changelog has a Deprecated section**

  D10 SIGNAL

  Announced this train: subject, replacement, removal version and date. Removed this train: under Removed.

  Evidence: Field Guide 08's `CHANGELOG.template.md`.

  Keep a Changelog 1.1.0 Deprecated and Removed categories. S13.

- [ ] **Retired packages carry an npm deprecation**

  D11 SIGNAL

  When a whole package is retired, `npm deprecate` puts the message in front of everyone who installs it. Only package owners can run it; an empty message undoes it.

  Evidence: The notice generator adds the command for `kind: "package"` entries.

  S04.

Put the notice where the code is written, not where the newsletter is sent.

SECTION 03

Hold the dates in CI.

A policy with dates is only as good as the day after the date. The checker runs on every pull request and on a daily schedule.

- [ ] **At least six months of notice, in calendar months**

  D12 CHECK

  Announced 13 October 2026, removable from 13 April 2027. Deprecating `color.link` in June for the October major window gives four months; the checker names 9 December as the earliest date; the next major window is 13 April 2027.

  Evidence: Rule E5, with `policy.minimumNoticeMonths`.

  Essay: minimum six months. Kubernetes: nine months or three releases for beta APIs. S03.

- [ ] **Removal only in a major, and only in a major window**

  D13 CHECK SEMVER

  `removalVersion` is a major above the announcing one, and `removalDate` is an April or October release date from the calendar.

  Evidence: Rules E4 and E6. The registry's `majorWindows` equal the major release dates Field Guide 08 prints for 2026 and 2027; a test checks it.

  SemVer rule 8. S01.

- [ ] **An overdue removal fails CI**

  D14 CHECK

  If the date passes and the API still ships, the build goes red until someone removes it or announces a new date.

  Evidence: Rule E8. On 14 April 2027 the kit's registry reports 3 errors, one per active entry, unless 5.0.0 has shipped.

  Kit rule.

- [ ] **Nothing is removed early**

  D15 CHECK

  The date is a promise in both directions. Removing ahead of it breaks teams who planned to the day you gave them.

  Evidence: Rule E10.

  Kit rule.

- [ ] **A date moves only by a new announcement**

  D16 PRACTICE

  Extend to the next major window, update the entry, and announce the new date through the same channels as the first. Never let it slip quietly.

  Evidence: The registry diff and a changelog line in the train that moves it.

  Kit rule.

The date is the policy. CI is how you keep it.

SECTION 04

Help teams get off it.

Remove when almost nobody is left. Usage can delay a date, never advance it.

- [ ] **Measure usage per consuming repository**

  D17 REGISTRY

  Consumers, how many still use it, and when measured. `Button` `href`: 9 of 41 repositories.

  Evidence: The optional `usage` block; a scheduled code search fills it.

  Essay: track adoption of the replacement.

- [ ] **High usage near the date needs a decision**

  D18 CHECK

  Within 60 days of removal and above 5% of consumers, the checker warns: help the rest migrate, or move the date and announce it.

  Evidence: Warning on the violations registry: 5 of 41 consumers (12.2%) with 19 days left.

  Essay: remove only under a threshold, for example 5% of consumers.

- [ ] **A codemod for every mechanical change**

  D19 CODEMOD

  The kit's transform turns `<Button href>` into `<LinkButton href>` and fixes the import.

  Evidence: 2 of 2 fixtures pass with jscodeshift 17.4.0; the CLI run over 2 files reports 2 ok and 0 errors.

  Carbon ships codemods through @carbon/upgrade. S10, S11.

- [ ] **No codemod? Say why**

  D20 REGISTRY CHECK

  `Badge tone="new"` becomes info or success depending on context. Say so, and nobody waits for a codemod.

  Evidence: `codemod` path must exist, or `codemodReason` is required (rule E9).

  Kit rule.

- [ ] **The codemod reports what it cannot change**

  D21 CODEMOD

  A Button with spread props may carry `href`. The transform leaves it and prints the file and line.

  Evidence: `Toolbar.tsx:10 Button receives spread props; check for href by hand (DEP-0007).`

  Kit codemod.

- [ ] **Removal ships with the migration in the changeset**

  D22 PRACTICE

  The removing major repeats the codemod command and guide link, for the team that missed six months of warnings.

  Evidence: Field Guide 08's gate rule C4; the kit's 2027.04 changesets.

  Essay: the checklist item for the migration guide exists because of one bad week.

Measure who is left, then help them leave.

APPENDIX A

One entry, every date.

`deprecations.json` holds the policy and one entry per deprecation. `deprecations.schema.json` (JSON Schema 2020-12) validates it in editors and in the checker.

deprecations.json (policy and one entry)

```
{
  "policy": {
    "minimumNoticeMonths": 6, "usageThreshold": 0.05, "usageWarningDays": 60,
    "majorWindows": ["2026-04-14", "2026-10-13", "2027-04-13", "2027-10-12"]
  },
  "deprecations": [{
    "id": "DEP-0007", "package": "@acme/ui", "kind": "prop", "rfc": "RFC-0042",
    "subject": "Button `href` prop", "stage": "deprecated",
    "announced": "2026-10-13", "announcedIn": "4.8.0",
    "removalVersion": "5.0.0", "removalDate": "2027-04-13",
    "replacement": "LinkButton",
    "migrationGuide": "https://acme-ui.example/migrations/5.0#button-href",
    "codemod": "codemods/button-href-to-link-button.mjs",
    "usage": { "consumers": 41, "using": 9, "measuredOn": "2026-09-21" },
    "owner": "@acme/design-system-core"
  }]
}
```

Fields regrouped for print. The kit's registry has four entries; DEP-0003 is Field Guide 03's `color.link` token, with the same replacement and removal version.

Rule	Fails when
E1	An entry does not match the schema: missing field, bad id, date or version format
E2	Two entries share an id
E3	Announced in a patch release
E4	Removal version is not a major above the announcing major
E5	Notice is shorter than `minimumNoticeMonths`
E6	Removal date is not one of the `majorWindows`
E7	No replacement and no reason, or no migration guide
E8	The removal date has passed and the entry is still `deprecated`
E9	The codemod file is missing, or there is no codemod and no reason
E10	Marked `removed` before its removal date

APPENDIX B

What the checker and the codemod print.

Both registries were checked with `--today 2026-09-24`. The codemod ran in a scratch install of jscodeshift 17.4.0.

terminal

```
$ node scripts/check-deprecations.mjs deprecations.json --today 2026-09-24
4 deprecations (3 active) checked on 2026-09-24; minimum notice 6 months.
  DEP-0003  @acme/tokens  `color.link` token: removal in 201 days, 3.0.0 on 2027-04-13
  DEP-0005  @acme/ui  Badge `tone="new"`: removal in 201 days, 5.0.0 on 2027-04-13
  DEP-0007  @acme/ui  Button `href` prop: announces in 19 days, 5.0.0 on 2027-04-13
deprecations: OK, 0 warning(s)

$ node scripts/check-deprecations.mjs examples/deprecations.violations.json --today 2026-09-24
  error    DEP-0002  E8  overdue: removal date 2026-04-14 passed 163 days ago ...
  error    DEP-0004  E5  4 months of notice (2026-06-09 to 2026-10-13); the policy
                         minimum is 6, so the earliest removal is 2026-12-09
  error    DEP-0006  E3  announced in 4.7.1, a patch release
  error    DEP-0008  E4  removal version 4.9.0 must be a major above 4.x
  error    DEP-0008  E2  duplicate id; DEP ids are never reused
  error    DEP-0008  E6  removal date 2027-02-09 is not a major window; the first
                         valid one is 2027-04-13
  error    DEP-0010  E7  no replacement and no replacementReason
  error    DEP-0010  E7  no migration guide; publish it on the day of the announcement
  error    DEP-0010  E9  codemod codemods/carousel-to-grid.mjs does not exist
  error    DEP-0011  E10 marked removed 201 days before its removal date 2027-04-13
  error    DEP-12    E1  (4 errors) owner required; id, kind and removalVersion malformed
  warning  DEP-0004  removal in 19 days but 5 of 41 consumers (12.2%) still use it
deprecations: 14 error(s), 1 warning(s)
```

Output from the kit. The violations listing is shortened for print: long messages wrapped, the active-entry summary omitted and DEP-12's four schema errors on one line.

codemods/__testfixtures__ (input, then output)

```
import { Button, Stack } from "@acme/ui";
      <Button tone="secondary" href="/settings" type="button">
        Settings
      </Button>
      <Button {...rest}>More</Button>

import { Button, Stack, LinkButton } from "@acme/ui";
      <LinkButton tone="secondary" href="/settings">
        Settings
      </LinkButton>
      <Button {...rest}>More</Button>   // reported, not changed
```

Excerpt. The second fixture shows an aliased import (`Button as UIButton`) replaced entirely, because nothing else uses it.

KEEP WITH THE ENTRY

Leave a deprecation record.

One record per deprecation, from the day it is proposed to the day it is gone. The registry holds the dates; this holds the reasons.

DEP id / subject / package	**RFC (link)**
Replacement and the release it shipped in	**Announced (version, date)**
Removal (version, date)	**Migration guide / codemod (links)**
Usage at announcement / at 60 days	**Date moved? New date and announcement (link)**
Removed in (pull request)	**Owner**

> **Never reuse the id.**
> A revoked or removed deprecation keeps its DEP number forever. Old changelogs, old warnings and old agent transcripts still point at it.

SOURCES / MAINTENANCE

Keep the guide current.

Sources checked 24 September 2026. The Polaris React repository (S09) is archived; its deprecation guidelines are cited as a published practice, not a current product.

S01 / Semantic Versioning 2.0.0
Rule 7: deprecation requires a minor; rule 8: breaking changes a major; FAQ on deprecating functionality.
https://semver.org/

S02 / Node.js, deprecated APIs
Documentation-only, application, runtime and end-of-life deprecations; identifiers kept when a deprecation is reversed.
https://nodejs.org/api/deprecations.html

S03 / Kubernetes deprecation policy
Removal only by a new API version; beta APIs nine months or three releases; deprecation warnings in responses.
https://kubernetes.io/docs/reference/using-api/deprecation-policy/

S04 / npm, npm deprecate
Deprecation messages on a version range; owners only; an empty message un-deprecates.
https://docs.npmjs.com/cli/v11/commands/npm-deprecate

S05 / TypeScript 4.0 release notes, @deprecated
Editors surface @deprecated in completions and as a suggestion diagnostic, typically struck through.
https://www.typescriptlang.org/docs/handbook/release-notes/typescript-4-0.html

S06 / typescript-eslint, no-deprecated
Reports uses of @deprecated code; requires type information.
https://typescript-eslint.io/rules/no-deprecated/

S07 / Primer, component lifecycle
Deprecated stage: documented alternatives and a warning shown to consumers.
https://primer.style/design/guides/component-lifecycle/

S08 / Atlassian Design System, release phases
Intent to deprecate and deprecated; no GA feature removed without a clearly announced period.
https://atlassian.design/release-phases

S09 / Polaris React, deprecation guidelines
Deprecations announced a month before a major; JSDoc @deprecated, dev warnings, automated and manual migration steps. Repository archived.
https://github.com/Shopify/polaris-react/blob/main/documentation/Deprecation%20guidelines.md

S10 / Carbon, v11 migration guide
Codemods through @carbon/upgrade for mechanical changes such as import paths and size props.
https://github.com/carbon-design-system/carbon/blob/main/docs/migration/v11.md

S11 / jscodeshift
Codemod runner and transform API; --dry, --print and the tsx parser; test utilities.
https://github.com/facebook/jscodeshift

S12 / React, versioning policy
Warnings added ahead of breaking changes; development warnings do not change production behavior.
https://react.dev/community/versioning-policy

S13 / Keep a Changelog 1.1.0
Deprecated and Removed as change types.
https://keepachangelog.com/en/1.1.0/

Maintenance: update `majorWindows` each autumn from Field Guide 08's calendar, remeasure usage monthly for entries within six months of removal, and rerun the codemod fixtures after a jscodeshift major. Update the PDF, HTML, Markdown and JSON together.