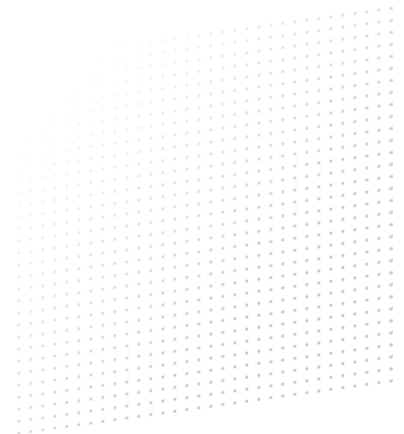

A RESOURCE FOR DESIGN SYSTEM AND PLATFORM TEAMS

Monthly release cadence.



A release train other teams can plan around.

Same day every month. Breaking changes twice a year.

22

items, each with its evidence

12

trains a year, on the second
Tuesday

02

major windows: April and
October

Make the date boring, so the content can change.

The essay behind this guide argues for monthly releases: weekly is too fast for product teams to absorb, quarterly too slow for bug fixes, and a fixed week lets every team plan around the release instead of being surprised by it. Large systems ship more often than that. Primer releases minors and patches weekly and bundles breaking changes into a major twice a year [S06]; Carbon ships minors every two weeks and patches in between [S07].

The lesson from both is the same: what teams struggle to absorb is breaking change, not release frequency. So this guide fixes two things. Every release lands on a date computed from a rule, a year ahead, with a code freeze and an RFC cut-off in front of it. Major versions ship only in two of those trains.

The kit prints the 2027 calendar and a 36-event .ics file, gates changesets against the train they will ship in (clean on two example trains, 6 errors on a bad one) and ships a Changesets 3 config that versioned the example packages from 4.7.0 to 4.8.0 and then to 5.0.0.

Version 1.0 / Sources checked 24 September 2026

Field Guide 08 of the Design × AI series, with 07 and 09 from the essay Enterprise UX that ships. Verified 24 September 2026: Node 22.22, @changesets/cli 3.0.3, python icalendar 7.3.0 for the .ics check.

Practical guidance, not a standard. The holidays in the kit are Finnish; replace them with yours. Acme UI and its versions are a worked example. Prepared with AI assistance and edited by hand.

START HERE

One train, three dates.

A release train leaves on time whether or not your change is on it. Everything in this guide hangs off three dates per month, all computed from one config file.

Date	When (Acme UI)	What happens
Decision cut-off	5 working days before freeze	RFCs targeting this train are accepted, or they move to the next one (Field Guide 07).
Code freeze	5 working days before release	The release branch is cut. Only fixes for this train merge to it.
Release	Second Tuesday of the month	<code>changeset version</code> , review, publish, release notes.
Major window	The April and October trains	The only trains that may carry a major version.

Train 2026.10: cut-off 29 September, freeze 6 October, release 13 October 2026, a major window. Owners: the release owner runs Sections 01 and 03, contributors and maintainers Section 02, the design system lead Section 04.

Setting up a train

Section 01. Edit `release-train.json`, print the calendar, publish the .ics file. Change the rule once a year at most.

Getting a change on it

Section 02. Every package change carries a changeset; the gate says whether it may ship in this train.

Running release week

Section 03 and `RELEASE-CHECKLIST.md`. Freeze, verify, version, publish, write the notes.

Something is on fire

Section 04. Security fixes and regressions ship as patches outside the train, through the same review.

Label	Meaning
<code>CALENDAR</code>	Produced by <code>release-calendar.mjs</code> from <code>release-train.json</code> .
<code>GATE</code>	Enforced by <code>check-changesets.mjs</code> ; the evidence line names the rule code.
<code>SEMVER</code>	Traces to Semantic Versioning 2.0.0.
<code>CHANGESETS</code>	Uses Changesets 3; verified with <code>@changesets/cli 3.0.3</code> .
<code>PROCESS</code> / <code>PRACTICE</code>	A step that leaves a record, or a working method with a review signal.

SECTION 01

Compute the calendar from a rule.

A list of dates goes stale. A rule and a generator give the same answer every year.

☐ **Release on a rule, not a list**

C01 CALENDAR

"Second Tuesday" in `release-train.json`. The generator turns it into dates for any year.

Evidence: `node scripts/release-calendar.mjs 2027` prints 12 trains, 12 January to 14 December.

The essay: every release ships the same week, and teams plan around it. Chrome runs a fixed four-week train. S08.

☐ **Pick a rule that keeps the freeze in the same month**

C02 CALENDAR

The essay's example is the first Tuesday. Then every 2027 freeze falls in the previous month, January's on 28 December.

Evidence: 12 of 12 freezes in the previous month with `--rule "first tuesday"`; 0 of 12 on the second Tuesday.

Kit output.

☐ **Count the freeze in working days**

C03 CALENDAR

Five working days, skipping weekends and the config's holidays. In calendar days, every holiday would quietly shorten stabilisation.

Evidence: In 2027 holidays stretch 2 of 12 freezes to 8 calendar days (January, May).

Kit output.

☐ **A release on a holiday moves forward, never back**

C04 CALENDAR

Moving back shortens the freeze; moving forward keeps it. The generator notes every move.

Evidence: With the kit's holidays, no 2026 or 2027 release moves.

Kit rule.

☐ **Publish the year as a calendar file**

C05 CALENDAR

One .ics file with an all-day event for every cut-off, freeze and release. Teams subscribe once.

Evidence: 36 events that parse in python icalendar 7.3.0: folded CRLF lines, stable UUIDs.

RFC 5545 sections 3.1 and 3.8.2. S10.

☐ **Name trains by the calendar, version packages by SemVer**

C06 SEMVER PRACTICE

Train `2027.04` is a CalVer label people can say. Packages keep SemVer: consumers depend on ranges.

Evidence: Release notes are titled by train; `package.json` versions stay `x.y.z`.

CalVer suits schedules; SemVer suits a public API. S01, S03.

The train leaves on time. Changes catch it or wait for the next one.

SECTION 02

Decide what may ride which train.

Minor and patch changes ride every train. Majors ride two a year.

☐ **Every package change carries a changeset**

C07 **CHANGESETS**

One file per change: packages, bump, a summary for consumers. Primer requires one on every API change.

Evidence: `changeset status` before the freeze; the kit's 2026.10 set gives two minor bumps.

S04, S06.

☐ **The bump follows the consumer's contract**

C08 **SEMVER**

Fix: patch. Compatible addition: minor. Breaks a consumer: major. `fix:`, `feat:` and `!` map the same way.

Evidence: Reviewers check the changeset bump against the diff.

SemVer rules 6 to 8; Conventional Commits 1.0.0. S01, S02.

☐ **Majors ship only in the April and October trains**

C09 **GATE**

Two known windows let product teams budget migration work. A major in any other train fails the gate.

Evidence: Rule C3: `major bump (@acme/ui) in 2026.11, which is not a major window`.

Primer bundles breaking changes twice a year. S06.

☐ **A major carries its migration in the changeset**

C10 **GATE**

A `Migration:` paragraph or a guide link. The essay's reason: a break without a guide once drew bugs from 47 teams in 48 hours.

Evidence: Rule C4. The kit's 2027.04 set passes; the violations set fails.

Essay, attributed. Kit rule.

☐ **A deprecation ships in a minor, with its DEP id**

C11 **GATE** **SEMVER**

Deprecating public API requires at least a minor. The DEP id links the changelog line to the registry entry and its removal date.

Evidence: Rule C5 (twice on the violations set: a patch, and no id).

SemVer rule 7. S01. Registry: Field Guide 09.

☐ **No placeholder changesets**

C12 **GATE**

"WIP" becomes a changelog line product teams read. Say what changed for them.

Evidence: Rule C2.

Changelogs are for humans, not machines. S09.

Release often. Break rarely, and on a date people already know.

SECTION 03

Run release week from a checklist.

Each checklist line exists because someone learned the hard way what happens without it.

☐ **RFCs are decided before the cut-off**C13 **CALENDAR** **PROCESS**

Undecided at the cut-off means the next train. RFC-0042: decided 10 September, cut-off 29 September.

Evidence: The calendar's decide-by date; the RFC's `decided` and `target` fields (Field Guide 07).

Kit rule.

☐ **The freeze cuts a release branch**C14 **PROCESS**

`release/2026.10` from `main` on freeze day. Fixes merge to the branch; features keep landing on `main`.

Evidence: The branch exists from the freeze date.

Recommended practice.

☐ **Verification leaves links, not ticks**C15 **PRACTICE**

Tests, axe, visual review, docs, migration guides: each with the run that proves it.

Evidence: `RELEASE-CHECKLIST.md` in the tracking issue, every line linked.

The essay's `ReleaseCheckList` type, turned into steps.

☐ **One product team installs the release candidate**C16 **PRACTICE**

The component can work while the upgrade path does not. A real consumer finds that first.

Evidence: Their upgrade pull request is linked in the tracking issue.

Essay: "The component worked. The upgrade path did not."

☐ **Release day is mechanical**C17 **CHANGESETS**

`changeset version` bumps and writes changelogs, a person edits them, `changeset publish` releases and tags. The kit's 2026.10 run: `@acme/ui 4.7.0 to 4.8.0`, `@acme/tokens 2.5.0 to 2.6.0`.

Evidence: Version pull request and publish log.

Verified with `@changesets/cli 3.0.3`. S04, S05.

☐ **Write release notes per train, for people**C18 **PRACTICE**

Generated changelogs are the input. Train notes are the output: highlights, breaks, deprecations with dates, fixes.

Evidence: `CHANGELOG.template.md`, posted on release day.

Keep a Changelog 1.1.0 categories. S09.

The checklist is the memory of every release that went wrong.

SECTION 04

Handle what cannot wait.

The essay promises no urgent hotfixes that bypass review. Keep the review. Drop the wait, for the two cases that justify it.

☐ Security fixes and regressions ship out of train

C19 **SEMVER** **PROCESS**

As a patch, as soon as they are verified. Everything else waits for the next train. Carbon ships patch releases between its minors for exactly this.

Evidence: Out-of-train patches are listed in the next train's Security section.

S07.

☐ Out of train never means out of review

C20 **PROCESS**

Same tests, same reviewer, same changeset. The only thing skipped is the calendar.

Evidence: The patch pull request has the same checks as a train release.

Essay: no hotfixes that bypass review.

☐ State how long the previous major gets fixes

C21 **PRACTICE**

Consumers who cannot upgrade in April need to know until when 4.x gets security patches. Carbon publishes active and maintenance phases per major.

Evidence: A support table in the docs with an end date per major.

S07.

☐ An empty train is announced, not discovered

C22 **CHANGESETS**

In Changesets 3, `changeset version` exits with code 1 when nothing is queued. Let the release job fail loudly, then post that the train carried nothing.

Evidence: Kit run: "No unreleased changesets found", exit code 1.

Changesets 3.0.0 changelog. S05.

Urgent changes skip the calendar. They never skip the review.

APPENDIX A

The 2027 train, as the kit prints it.

`release-train.json` holds the rule, the freeze length, the major months and the holidays. The generator does the rest.

terminal

```
$ node scripts/release-calendar.mjs 2027
Acme UI release train: second tuesday, freeze 5 working days earlier
Train    Decide by    Freeze      Release     Window  Notes
2027.01  2026-12-23   2027-01-04   2027-01-12  minor   holidays: freeze spans 8 days
2027.02  2027-01-26   2027-02-02   2027-02-09  minor
2027.03  2027-02-23   2027-03-02   2027-03-09  minor
2027.04  2027-03-30   2027-04-06   2027-04-13  major
2027.05  2027-04-26   2027-05-03   2027-05-11  minor   holidays: freeze spans 8 days
2027.06  2027-05-25   2027-06-01   2027-06-08  minor
2027.07  2027-06-29   2027-07-06   2027-07-13  minor
2027.08  2027-07-27   2027-08-03   2027-08-10  minor
2027.09  2027-08-31   2027-09-07   2027-09-14  minor
2027.10  2027-09-28   2027-10-05   2027-10-12  major
2027.11  2027-10-26   2027-11-02   2027-11-09  minor
2027.12  2027-11-29   2027-12-07   2027-12-14  minor
12 trains, 2 major windows, 0 freezes in the previous month, 2 freezes stretched by holidays, 0 releases moved.
```

examples/acme-ui-2027.ics (one event)

```
BEGIN:VEVENT
UID:2027.04-release@acme-ui.example
DTSTAMP:20260924T000000Z
DTSTART;VALUE=DATE:20270413
DTEND;VALUE=DATE:20270414
SUMMARY:Acme UI release train 2027.04: release (major window)
DESCRIPTION:Major versions may ship in this train.
TRANSP:TRANSPARENT
END:VEVENT
```

`TRANSP:TRANSPARENT` keeps the all-day events from showing people as busy.

APPENDIX B

The gate, the config and what Changesets wrote.

The gate reads the train from the same calendar, so it cannot disagree with the dates you published.

terminal

```
$ node scripts/check-changesets.mjs examples/changesets-2026.10 --train 2026.10
Train 2026.10: freeze 2026-10-06, release 2026-10-13, major window.
4 changesets: 0 major, 3 minor, 1 patch.
changesets: OK

$ node scripts/check-changesets.mjs examples/changesets-violations --train 2026.11
Train 2026.11: freeze 2026-11-03, release 2026-11-10, minor and patch only.
4 changesets: 1 major, 1 minor, 1 patch.
  button-elevated.md C3 major bump (@acme/ui) in 2026.11, which is not a major window
  button-elevated.md C4 major bump without a Migration: paragraph or a link to the migration
guide
  deprecate-tooltip-delay.md C5 a deprecation is at least a minor release (SemVer 2.0.0 rule
7)
  deprecate-tooltip-delay.md C5 a deprecation must cite its DEP-NNNN registry id
  rename-space-scale.md C1 needs "package": major|minor|patch lines (cannot read:
"@acme/tokens": breaking)
  wip.md C2 summary is empty or a placeholder; write what changed for the consumer
changesets: 6 error(s)
```

The 2027.04 set, with two majors in a major window, also passes.

.changeset/config.json

```
{
  "$schema": "https://unpkg.com/@changesets/config@4.0.1/schema.json",
  "changelog": "@changesets/cli/changelog",
  "commit": false,
  "fixed": [],
  "linked": [],
  "access": "public",
  "baseBranch": "main",
  "updateInternalDependencies": "patch",
  "ignore": [],
  "privatePackages": false
}
```

Changesets 3 removed the `prettier` option (use `format`) and no longer versions private packages by default [S05]. Swap in `@changesets/changelog-github` for pull request links; it needs a GitHub token, so the kit's offline run used the default.

packages/ui/CHANGELOG.md (generated by changeset version)

4.8.0

Minor Changes

- 3390aec: Add `LinkButton`: navigation that looks like a button and renders an ``. Accepted in RFC-0042.
- 3390aec: Deprecate Button `href` prop (DEP-0007). Use LinkButton instead. Removed in 5.0.0 on 13 April 2027. ...

Patch Changes

- 3390aec: Fix the `Button` focus ring being clipped inside containers with `overflow: hidden`.
- Updated dependencies [3390aec]
 - @acme/tokens@2.6.0

The deprecation line is the changeset Field Guide 09's notice generator writes. Running the 2027.04 set next produced @acme/ui 5.0.0 and @acme/tokens 3.0.0.

KEEP WITH THE TRAIN

Leave a release record.

One record per train, in the tracking issue. It is what the next release owner reads when this train's decisions come back as questions.

Train / release date / window	Release owner / backup
Changeset gate run (link)	Versions published
RFCs shipped (ids)	Deprecations announced / removed (DEP ids)
Verification runs (links)	Canary consumer upgrade (link)
Out-of-train patches since last train	Slipped to next train, and why

A missed train is information.

Write down what slipped and why. Three trains in a row missing the same kind of change is a process problem, not a scheduling one.

SOURCES / MAINTENANCE

Keep the guide current.

Sources checked 24 September 2026. Release cadences are quoted from each system's own contributor documentation on that date and change over time; recheck them before you cite them to your organisation.

S01 / Semantic Versioning 2.0.0

Rules 6 to 8 for patch, minor and major; rule 7: deprecation requires a minor.

<https://semver.org/>

S02 / Conventional Commits 1.0.0

fix maps to PATCH, feat to MINOR, BREAKING CHANGE or ! to MAJOR.

<https://www.conventionalcommits.org/en/v1.0.0/>

S03 / Calendar Versioning

YYYY and 0M tokens; when calendar-based versions fit better than semantic ones.

<https://calver.org/>

S04 / Changesets, config file options

changelog, commit, fixed, linked, access, baseBranch, updateInternalDependencies, ignore, privatePackages.

<https://github.com/changesets/changesets/blob/main/docs/config-file-options.md>

S05 / Changesets CLI changelog, 3.0.0 to 3.0.3

prettier replaced by format; private packages not versioned by default; version exits 1 with nothing to release.

<https://github.com/changesets/changesets/blob/main/packages/cli/CHANGELOG.md>

S06 / Primer React, CONTRIBUTING

Minor and patch changes weekly; breaking changes bundled into a major twice a year; changesets required.

<https://github.com/primer/react/blob/main/contributor-docs/CONTRIBUTING.md>

S07 / Carbon, release schedule

Biweekly minors in the active phase; patches between minors; maintenance phase for security and critical fixes.

<https://github.com/carbon-design-system/carbon/blob/main/docs/release-schedule.md>

S08 / Chrome, a faster release cycle

Six weeks to four weeks from Chrome 94 (2021): a fixed train shortened for faster fixes, not more features.

<https://developer.chrome.com/blog/faster-release-cycle>

S09 / Keep a Changelog 1.1.0

Added, Changed, Deprecated, Removed, Fixed, Security; changelogs are for humans.

<https://keepachangelog.com/en/1.1.0/>

S10 / RFC 5545, iCalendar

CRLF content lines folded at 75 octets; PRODID, VERSION, UID and DTSTAMP required; DTEND non-inclusive.

<https://www.rfc-editor.org/rfc/rfc5545>

Maintenance: regenerate the calendar and the .ics file each autumn for the next year, after updating the holidays. Recheck the Changesets config at each Changesets major and the cadences in S06 and S07 yearly. Update the PDF, HTML, Markdown and JSON together.