Sleep Better on Release Day

A Modern Testing Strategy for JavaScript SDKs & Components

CKEditor

~15 years ago, my every Thursdays meant clicking every button the app on specific browser and .NET version by hand to catch regressions before release

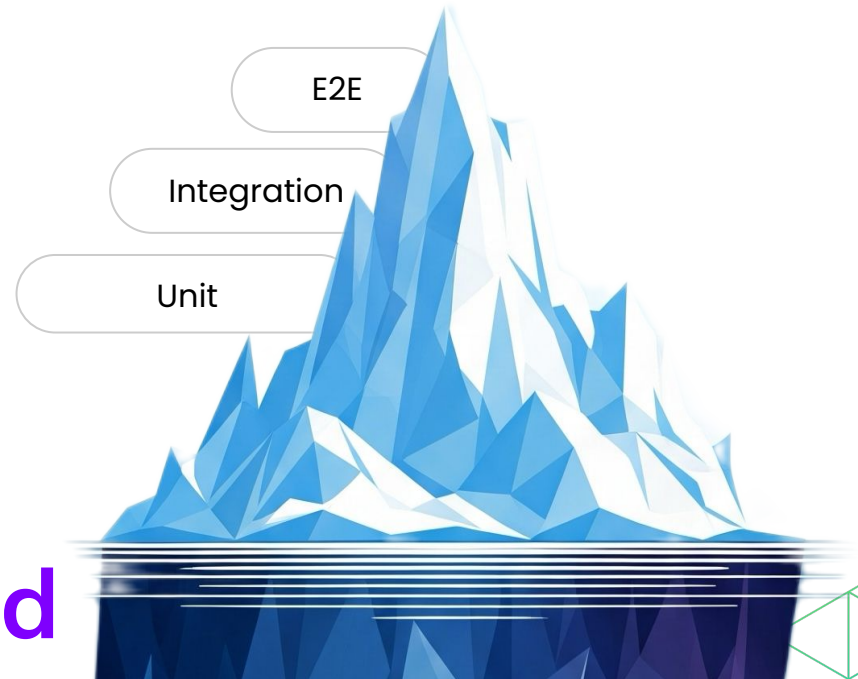
Ondřej Chrastina

Developer Advocate

CKEditor

ondrej.chrastina.dev

Testing pyramid



Testing iceberg

qa-e2e-tests repository

Scale & Versatility

- Massive suite of **40k tests** ensuring comprehensive coverage.
- Runs across various product scopes, from single components to full systems.
- Multipurpose utility: used for regression, validation, and release readiness.

The Challenge

- The sheer volume and diversity of tests reveal **various types of problems**.
- High complexity requires sophisticated triage and maintenance.

Modus Operandi

Repo
& Runners

XP
Experience

6
Interviews

Kacper
Tomporowski

Kamil
Piechaczek

Łukasz
Zając

Martyna
Buczyńska

Karolina
Schild

Mateusz
Bagiński

AI
Agent

"Since we were improving our test base and finding more stuff, other teams starting to reach to us."

— **Kacper Tomporowski**

The 3 Pillars

1. Integration Matrix

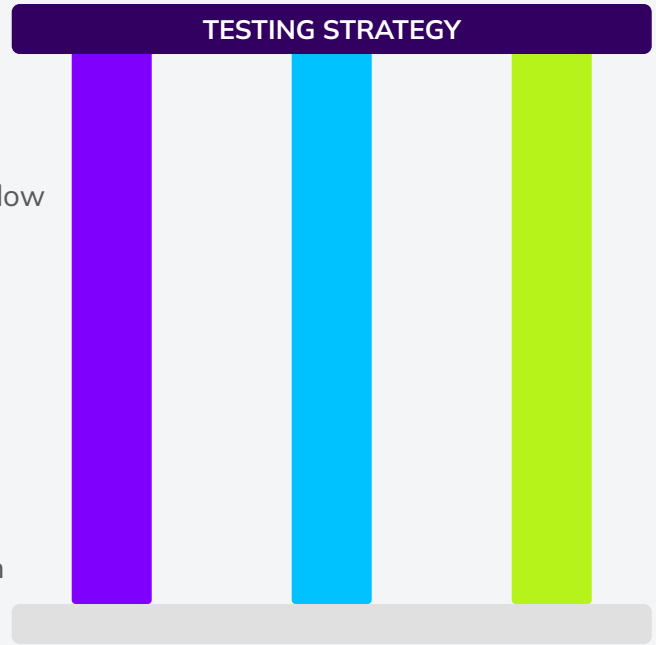
Comprehensive mapping of system intersections and data flow dependencies.

2. Timeline Strategy

Optimized release scheduling with automated regression checkpoints.

3. AI Models Interoperability

Seamless cross-functional testing using advanced AI-driven verification agents.



Integration Matrix

Pillar 1

The Integration Matrix

Frameworks	React | Vue | Angular | Next.js | Vanilla JS
Bundlers & Delivery	Vite | webpack v5 | esbuild | CDN | UMD | ESM
Browser Engines	Chrome (Blink) | Firefox (Gecko) | Safari (WebKit) | Edge (Blink)
Renderers	Browser editor | PDF export

"We use real browsers, real interactions. We don't use editor API for interacting with the editor. We try to simulate real typing, real mouse clicks."

— Kacper Tomporowski

Broad vs. Deep

Broad Coverage

Every sample in the matrix → Sanity checks only (fast, catches regressions)

Deep Validation

Default sample (Vite + Vanilla JS) → Full feature tests (slow, catches bugs)

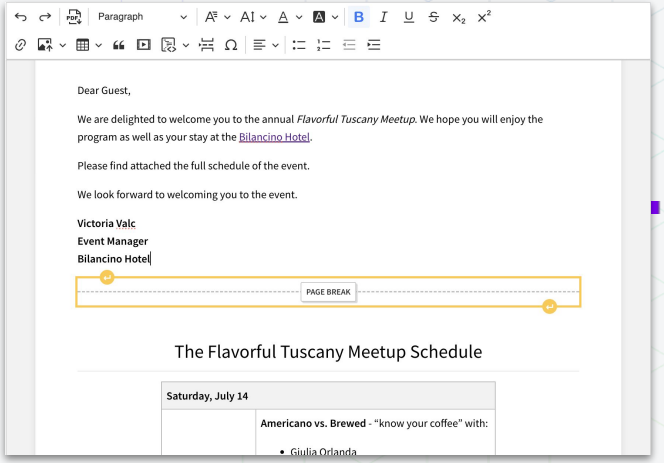
"Broad where it matters. Deep where it's safe."

"It's plain with full editor feature set, but doesn't have any environment customizations, not dedicated to any specific feature or framework."

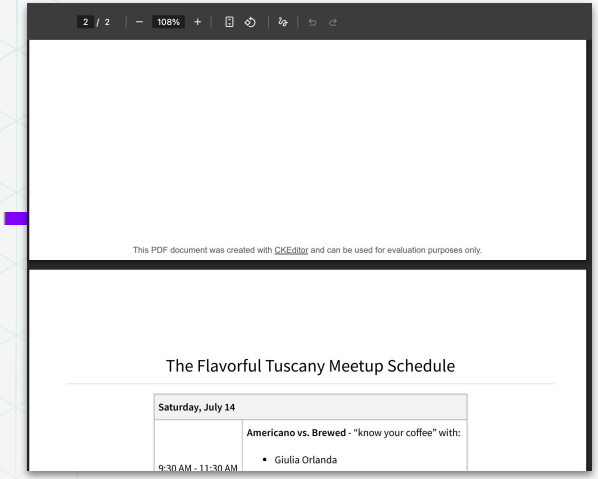
— **Karolina Schild** (on why Vite + Vanilla JS is the default sample)

Two Renderers, One Truth

Browser



PDF export



"Unit tests can't read a PDF. Visual regression can — sometimes."

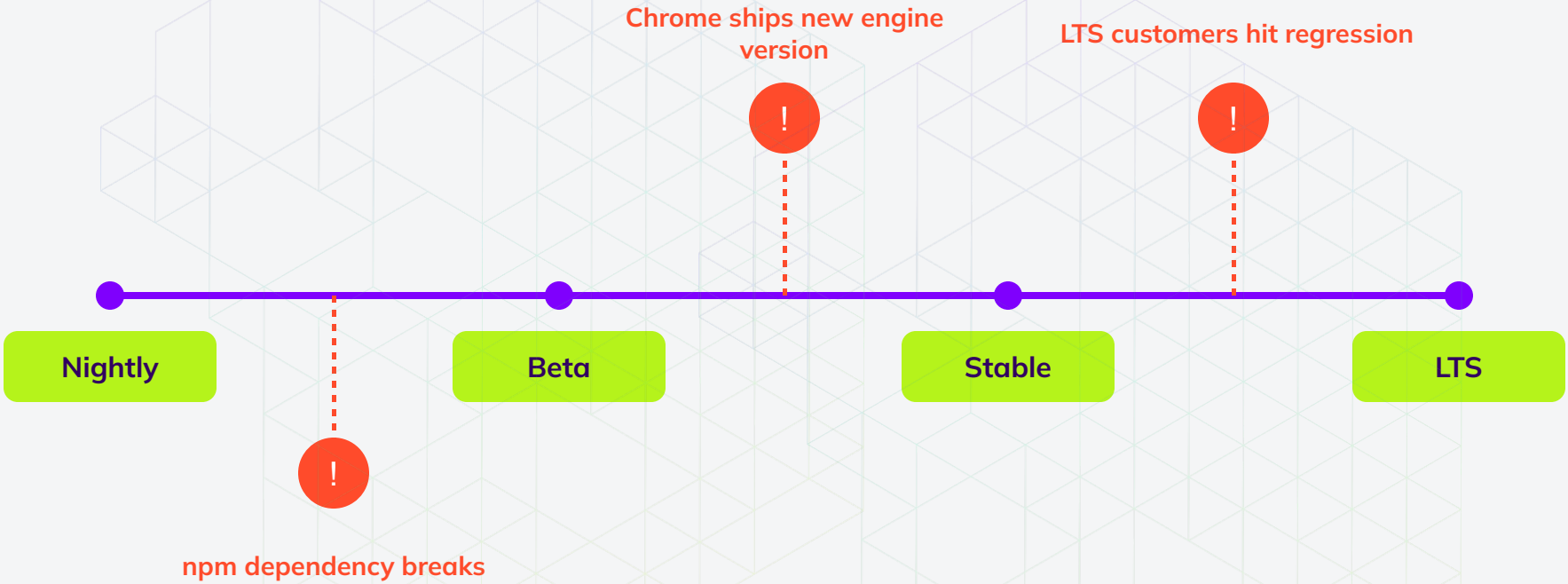
"Compare export PDF documents that used to work correctly with pagination with these ones on recent main branch."

— Mateusz Bagiński)

Timeline Strategy

Pillar 2

You Don't Control the Timeline



Reactive: Chasing the Browser

Chrome 146 (Before)

```
border-right-style: none
```

Chrome 147 (After)

```
border-right: medium none
currentcolor
```

Fri 9:19 AM

Ticket Opened

Fri 1:30 PM

Fix Merged

"Paste from Office: Same rendering. Different string."

Proactive: Forecasting the Browser

Caught: Chrome 143 spec change

```
— 3 weeks before stable release
```

In-code browser checklist

```
[x] Chrome Beta
[ ] Firefox
[ ] Safari
```

2025-10-30

Insight Identified

"Your tests should know what's coming, not just what is."

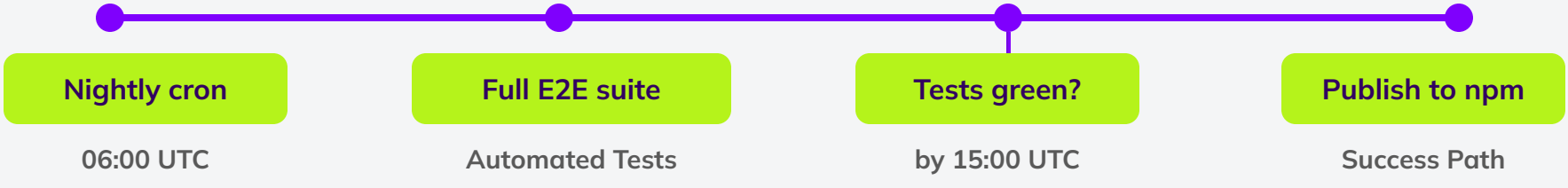
"CKEditor is released on Wednesday, the same day as Chrome updates. It's even crazier when you think that we are perfectly matched with each other."

— Kamil Piechaczek

Gate the Release

Condition Check

```
if (tests !== 'green') {
  block_release();
}
```



"If the suite isn't green, the release doesn't ship."

AI Models Interoperability

Pillar 3

How Do You Test a Maybe?

Deterministic Code

Predictable and rigid logic paths.

```
input A → output B
assert(output === B)
```

Result: **Same input → Same output**

Non-deterministic AI

Probabilistic and fluid responses.

```
input A → output B, C, or D
assert(intent === "B")
```

Result: **Same input → Different output**

Make it deterministic

Problem: Test fails because AI rewrites the same sentence differently each run.

Solution: Record a real AI response once → replay it in tests.

Real AI call — Record response — Fixture file — Replay in CI — Stable snapshot

"Visual regression can't handle non-deterministic text."

"Mocking only when necessary is the strategy that we have."
— **Łukasz Zając**

Precision is Stability

Vague Prompt = Flaky Result

User: "Add paragraph numbers."

- `Run 1: #1, #2`
- `Run 2: I., II.`
- `Run 3: 1), 2)`

Result: Visual regression fails due to format drift.

Precise Prompt = Fixed Contract

User: "Add the word 'Paragraph' followed by its number and a dot..."

- `Run 1: Paragraph 1.`
- `Run 2: Paragraph 1.`
- `Run 3: Paragraph 1.`

Result: One format, zero ambiguity, test passes.

Bonus Strategy: Seed a deliberate typo in the first heading → The "Review" feature should catch it every time to verify functional logic.

"Your prompt is your test setup. Ambiguity is flakiness."

"Your prompt is your test setup. Ambiguity is flakiness."
— **Łukasz Zając**

Prompt Inception

The "Stale" Failure

User: `"Does CKEditor AI support web search?"`

AI Answer: **"No."**

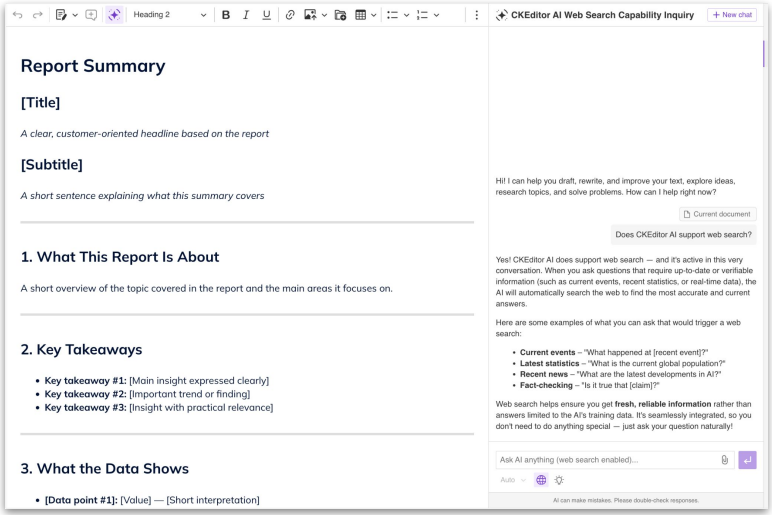
Issue: Model relies on internal training data cutoffs rather than current product features.

The "Contract" Fix

Updated Prompt:

`"Use the following documentation to answer: [Live Docs URL] ... Does CKEditor AI support web search?"`

AI Answer: **"Yes, via the search tool."**



Use AI to Test AI

Layer 1: Deterministic

FAST & PROGRAMMATIC

- HTML output is valid
- Correct tool was called
- Response type is modification, not chat

Layer 2: LLM-as-judge

SEMANTIC EVALUATION

- "Did it fix only the specified paragraph?"
- "Does the list contain appropriate items?"
- Separate judge model grades the response

Output: pass/fail rate per model → powers auto-mode and recommended-model selection

"When 'did it do the right thing?' is a semantic question, use another model to answer it."

Red CI ≠ Bug in Your Code

Error & Cause

React Hydration Error #418

The CI fails with a cryptic hydration mismatch `#418` in the console logs.

Cause: Cypress execution environment race condition — not reproducible in real browsers.

Resolution

Documented Suppression

- Identified as "Ghost in the Machine"
- Applied targeted lint/error suppression
- Defined revisit criteria for CI environment upgrade

"In testing, there is no one-size-fits-all strategy"
...expect the unexpected."

"We decided to keep the exception and schedule the revalidation once the fix is provided."

— Martyna Buczyńska

Sleep Better on Release Day

1. Integration Matrix

Test what makes sense, thinking beyond standard dimensions like PDF rendering.

2. Timeline Strategy

Proactively gate the timeline with quality checks, or the timeline will gate your release.

3. AI Interoperability

Use AI to test AI. Mock for stability; use LLM-as-judge for semantic evaluation.

Expect the unexpected. The system is watching.

CKEditor

Sleep Better on Release Day

Ondřej Chrastina
Developer Advocate
CKEditor

ondrej.chrastina.dev