

STANDARD OPERATING PROCEDURE · WW-009

Diagnose Core Web Vitals and prioritise fixes

Read real-user Core Web Vitals data for a site, identify which page templates fail and why, and produce a prioritised fix list a developer can act on.

TECHNICAL SEO

VERSION 1.0	EFFECTIVE 5 October 2026
NEXT REVIEW 5 April 2027	TIME 90 minutes
DIFFICULTY Advanced	RELATED WW-006
AUDIENCE SEO specialists and developers responsible for site performance	
TOOLS Google Search Console · PageSpeed Insights · Chrome DevTools	

Purpose

Core Web Vitals measure how fast a page loads its main content (Largest Contentful Paint, LCP), how quickly it responds to interaction (Interaction to Next Paint, INP) and how visually stable it is (Cumulative Layout Shift, CLS). This procedure uses real-user data to find which templates fail, then lab tools to find the cause, so engineering time goes to fixes that improve the experience of actual visitors. It is finished when every failing template has a diagnosed cause and a prioritised, owner-assigned fix.

Scope

In scope

- Reading the Search Console Core Web Vitals report
- Using PageSpeed Insights field and lab data per template
- Identifying the LCP element, layout shift sources and slow interactions
- Writing and prioritising fixes

Out of scope

- Implementing code changes (handled by the developer's own process)
- Hosting migrations and server tuning
- Accessibility and general UX audits

Roles and responsibilities

Role	Responsibility
Technical SEO	Gathers data, diagnoses causes and writes the fix list.
Developer	Validates feasibility, estimates effort and implements fixes.
Account manager	Agrees priorities and timelines with the client.

Prerequisites

- Search Console access for the property (WW-004).
- A list of the site's main page templates with one example URL each.
- Google Chrome with DevTools.

Procedure

Find failing templates from field data

Start from what real visitors experience.

1. In Search Console, open **Core Web Vitals** and review the **Mobile** report first, then **Desktop**.

2. For each issue listed as **Poor** or **Needs improvement**, open it and note the metric, the affected URL group and an example URL.
3. Map each URL group to a page template (for example, all blog posts or all product pages).

[!NOTE] The thresholds for **Good** at the 75th percentile of visits are LCP of 2.5 seconds or less, INP of 200 milliseconds or less, and CLS of 0.1 or less.

Analyse each failing template

Use one representative URL per template.

1. Run the example URL through [PageSpeed Insights](#). Record the field data (real users) at the top, then the lab score and diagnostics below.
2. For LCP, open the **Largest Contentful Paint element** diagnostic and note which element it is (usually a hero image or heading) and the breakdown of time to first byte, load delay, load time and render delay.
3. For CLS, open **Avoid large layout shifts** and note which elements move: commonly images without dimensions, late-loading banners, embeds or web fonts.
4. For INP, open **Minimise main-thread work** and **Reduce JavaScript execution time**, and note heavy scripts, especially third-party tags and chat widgets.

Confirm causes in Chrome DevTools

Verify the diagnosis before a developer spends time on it.

1. Open the URL in Chrome, open **DevTools**, then the **Performance** panel, and record a page load with mobile throttling enabled.
2. Confirm the LCP marker matches the element identified, and check whether its image is lazy-loaded, oversized or discovered late.
3. Interact with the main menu and a form, and look for long tasks over 50 milliseconds that coincide with the interaction.

Write the fix list

Give each fix a clear cause, action and expected impact.

1. For each finding, write a fix using the pattern **Template · Metric · Cause · Action**. Common actions:
 - **LCP**: serve the hero image in a modern format at the displayed size, do not lazy-load it, add `fetchpriority="high"`, and reduce server response time with caching.
 - **CLS**: set `width` and `height` on images and embeds, reserve space for banners, and use `font-display` with matched fallback fonts.
 - **INP**: defer or remove non-essential third-party scripts, break up long tasks, and load chat and tracking scripts after interaction where possible.
2. Score each fix by the share of traffic on that template and the size of the gap to **Good**, and agree an order with the developer.

Validate after release

Field data takes time to reflect fixes.

1. After fixes are deployed, re-run PageSpeed Insights lab tests to confirm the specific diagnostics are resolved.
2. In Search Console, open the issue and click **Validate fix**. Review the field data again after 28 days.

Quality checks

- ☐ Every Poor or Needs improvement issue in Search Console is mapped to a template.
- ☐ Each failing template has an identified LCP element, CLS sources or INP culprits as relevant.
- ☐ Every fix follows the Template · Metric · Cause · Action pattern with an owner.
- ☐ Validation has been started in Search Console after release.

Troubleshooting

Symptom	Likely cause	Fix
No field data in PageSpeed Insights	Not enough real-user traffic for the URL	Use origin-level data and the Search Console report instead.
Lab score good but field data poor	Real visitors use slower devices and networks	Test with mobile throttling and prioritise field data.
Validation fails	Not all URLs in the group were fixed, or traffic mix changed	Re-check a sample of affected URLs, then re-validate.

Records

- Save the fix list and before/after PageSpeed Insights reports in the client's technical folder.
- Record the validation start date and the 28-day review outcome.

References

- [Web Vitals \(web.dev\)](#)
- [Core Web Vitals report \(Search Console Help\)](#)
- [Optimise Largest Contentful Paint \(web.dev\)](#)
- [Optimise Interaction to Next Paint \(web.dev\)](#)
- [Optimise Cumulative Layout Shift \(web.dev\)](#)

WRIGHT WEBSITES · WW-009 · V1.0 · COVERED BY THE RESOURCES LICENCE AT
WRIGHTWEBSITES.CO/LEGAL/RESOURCES-LICENCE