



The Developer's Guide to Improving Core Web Vitals



[Try Raygun in AWS Marketplace](#)



Table of contents

CX in the age of infinite information	3
<hr/>	
Understanding Core Web Vitals	4
<hr/>	
The background	4
Defining Core Web Vitals	4
Calculating your score	5
Why you should care	7
<hr/>	
The opportunity	7
The outcome	7
How to improve your Core Web Vitals	8
<hr/>	
The target	9
The tactics	10
Improving LCP	10
Improving FID	11
Improving CLS	12
The workflow	13
<hr/>	
Step 1: Identify opportunities with RUM	13
Step 2: Prioritize optimizations	14
Step 3: Implement changes, learn, and iterate	15
Step 4: Monitor trends overtime	15
Core Web Vitals in Raygun	16
<hr/>	
Filter data by what matters to you most	16
Unlock greater context in order to take action	17
View trends over time and set up custom alerts	18
Taking control	19
<hr/>	

CX in the age of infinite information

User experience matters. Digital adoption has surged since the pandemic transformed everyday life, and while the market is growing at pace, customer expectations are sky-high and the competition is more cutthroat than ever.

Developers are on the front lines having to deliver increasingly rich and complex digital experiences while trying to ensure everything is as frictionless as possible. With more metrics available than ever before, it has become increasingly difficult for developers to know where to focus their efforts for the greatest impact. Google has stepped in to provide an answer to that question, standardizing the user experience with three Core Web Vitals (CWV).

In this definitive guide, you'll get best-practice advice, a proven workflow and actionable tips to start improving your Core Web Vitals and enhancing your end-user experience now.



Understanding Core Web Vitals

The background

In 2020, Google first introduced CWV as a subset of their well-established [Web Vitals](#) initiative to simplify the complex landscape of monitoring digital experiences. Each represents a distinct facet of the user experience: content loading speed, interactivity, and visual stability.

In May 2021 these were officially implemented as a ranking factor in the Google Search algorithm. This creates elevated pressure on businesses to ensure they're meeting the standard specified by Google, or risk losing organic placements, and in turn, diminishing business and revenue.

Defining CWV

In the interests of ranking pages that provide consistent quality, [Google has selected and defined](#) three key measures of user experience:



LCP (Largest Contentful Paint)

The amount of time to render the largest content element visible in the viewport, from when the user requests the URL. The largest element is typically an image or video, or perhaps a large block-level text element.

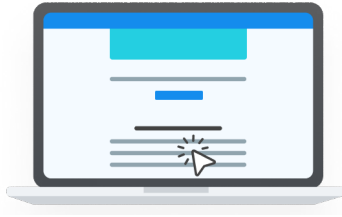
This is important because it tells the reader that the URL is actually loading.



FID (First Input Delay)

The time from when a user first interacts with your page (clicked a link, tapped on a button, and so on) to the time when the browser responds to that interaction. This measurement is taken from whatever interactive element the user first selects.

This is important to make the user feel that the page is responsive in contexts where they need to perform an action.



CLS (Cumulative Layout Shift)

CLS measures the proportion of the viewport affected and the distance shifted for the largest burst of unexpected layout shifts across the entire lifespan of the page. The score ranges from zero upwards, where zero means no shifting and the larger the number, the more shifts on the page.

This is important because page elements shifting while a user is trying to interact create a bad user experience.

Other web vitals

Core Web Vitals are your primary opportunity and your chief focus right now - but it's important to view them as a component of a larger set of Web Vitals metrics. Google sees these other metrics as less critical to user experience, but they can still help with understanding and addressing CWV issues and building a faster and higher-functioning site. You'll encounter terms like [First Contentful Paint \(FCP\)](#) and [Time to Interactive \(TTI\)](#) while using testing and diagnostic tools, and these are relevant as supplementary measures of the speed and smoothness of your page.

Calculating your score

If you've ever tried monitoring your Core Web Vitals on multiple tools, you may have noticed some variation in your scores. This is because the metrics are based on different datasets - synthetic/lab, CrUX, and Real User Monitoring (RUM), all of which serve a different purpose when monitoring and improving your scores.

Let's dive into what makes each source unique, the pros and cons of each, and which tools will surface which datasets.

Lab data

Lab data is perfect for benchmarking and testing site performance using a controlled environment and a standardised set of conditions. Popular Google Chrome tool, [Lighthouse](#), calculates your Core Web Vital scores by simulating a mobile device running on a fast 3G network and slowed CPU, while also identifying opportunities to improve the user experience.

Lab data can be invaluable for getting a quick state-of-play and understanding the impact of recent optimizations, and it's also ideal for testing a website before launch when real user data just doesn't exist yet. However, it can't be relied upon for a true indication of your users' experience under real-world conditions.



Field data

Field data is crucially important for understanding how real users are experiencing your website. Field data analyzes how real people interact with and experience your website out in the wild, across different browsers, devices, internet speeds and other variables, to give you more accurate insights into your true CWV scores than synthetic lab data.

These real user insights can be gathered in two distinct ways - from a [Real User Monitoring solution](#) such as Raygun, or from a tool that uses the CrUX dataset, such as [PageSpeed Insights](#).

- **Real User Monitoring (RUM)** will give valuable insights into how every user is experiencing your CWV, providing you a comprehensive view of your user's digital experience.
- **Chrome User Experience (CrUX)** is aggregated from users who are using Google Chrome, have opted-in to syncing their browsing history, have not set up a Sync passphrase, and have usage statistic reporting enabled, with metrics averaged out over the past 28 days.

While Google's tools provide enough data for you to have some sense of your CWV performance, RUM is key to understanding what's happening in real time across all your visitors. In fact, Google's CWV team recommends supplementing their workflow with RUM for accurate data and diagnostics.



“CrUX...does not provide the detailed, per-pageview telemetry that is often necessary to accurately diagnose, monitor, and quickly react to regressions. As a result, we strongly recommend that sites set up their own real-user monitoring.”

Philip Walton,
Google

Why you should care

The opportunity

Core Web Vitals are brand new, and they're not going anywhere anytime soon – there's nothing quite like the indisputable industry-leader weighing in on the importance of web performance. The position Google has taken has brought front-end performance directly into the boardroom, clearly connecting UX and business outcomes. Not only does this present a very real opportunity for developers to get the attention and investment they deserve, but also gives them widely understood metrics to focus on.

Surprisingly, your competitors likely haven't caught wind of their importance yet, and their lag is your (potential) gain. A March 2021 assessment of [500 prominent ecommerce sites](#) found that only 16.6% were achieving "Good" on all three metrics, so by reading this report you're already well ahead of the curve.

This window of opportunity won't last forever. By taking a proactive approach to improving your CWV scores, you not only gain an edge on your Google search rankings, but more importantly, lead the standard for what a great digital experience looks like.

The outcome

Focusing on these metrics can pull you ahead of the pack. In the initial two months after the update that added CWV to Google's search ranking criteria, domains that weren't meeting at least one of the [Core Web Vitals values](#) achieved [3.7% less Google visibility](#) than domains that met all three CWV standards.

As we know, search ranking isn't the only benefit – Google picked these particular metrics for good reason. Core Web Vitals have a significant bearing on both bounce rates and conversions with faster, smoother sites resulting in



Only 16%

of 500 prominent ecommerce sites are achieving a "Good" score across all three CWV metrics.



24%

of users are less likely to abandon pages mid-load when shopping sites meet the "Good" mark on all three metrics.

a healthier bottom line. Shopify reports that [57% of shoppers have left a slow site](#) to buy from a competitor. When shopping sites meet the “Good” mark on all three metrics, [users are 24% less likely to abandon the page mid-load](#).

Swappie

Swappie increased mobile revenue by 42% by focusing on improving CWV scores and reducing average page load time by 23%.

AliExpress™

AliExpress reduced their bounce rate by 15% after improving their CLS tenfold and LCP two fold.

NETZWELT

Netwelts achieved a 27% increase in page views and a 18% increase in ad revenue by boosting CWV scores across the board.

vodafone

Vodafone increased sales by 8% and their lead to visit rate by 15% after improving LCP by 31%

Core Web Vitals aren't just an arbitrary benchmark – they're the factors that make your site feel fast and seamless to your users and translate into real gains in revenue.



How to improve your Core Web Vitals

Google judges each of your Core Web Vitals independently, so it's important to spread your optimizations across all three. Simply focusing on one or two metrics will backfire in the long run, so aim to get all three hitting “Good” for the best results.

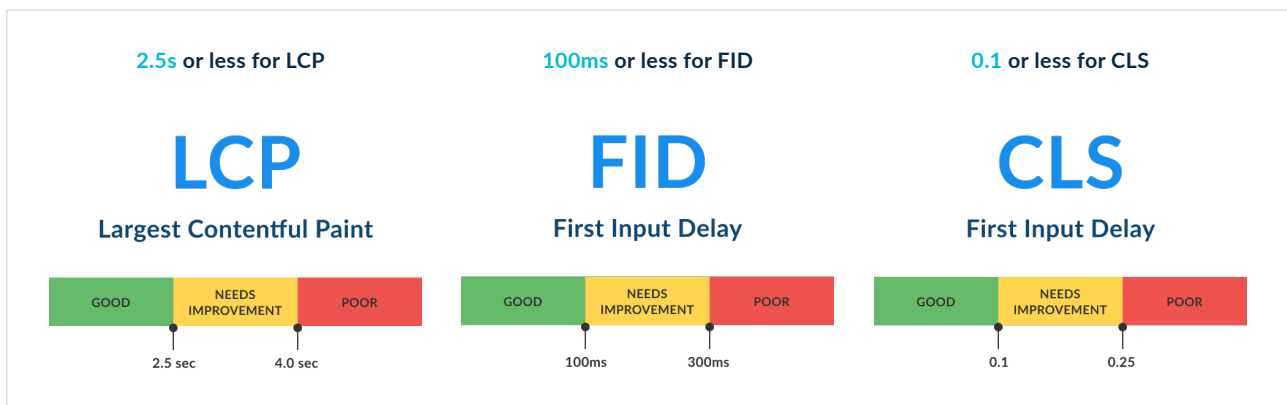
When it comes to organic search, Google will monitor the performance of individual pages, not your entire site. Don't get overwhelmed thinking you have to optimize every page and asset you've got - focus your attention on the pages that get the most web traffic and/or are of the greatest importance to your business.

The target

Regardless of what tool you use to measure Core Web Vitals, your scores are defined in the same way and divided into three distinct categories: “Good”, “Needs Improvement”, and “Poor”.

The Core Web Vitals are heavily focused on the 75th percentile of users for the preceding 28-day period. For example, to achieve a ‘Good’ LCP score, 75% of your users must experience an LCP of less than 2.5 seconds. If you don’t meet this threshold, but 75% of your users have an LCP less than 4 seconds, then you’ll fall into the ‘Needs work’ segment. And if you can’t hit the threshold of 75% of users hitting an LCP of less than 4 seconds, then (you guessed it) you fall into the ‘Poor’ category.

Google defines a “Good” experience when 75% or more of your users experience the following:



It’s worth noting that once you’ve met these thresholds for “Good”, your search ranking won’t be impacted by any gains above and beyond this. However, the other benefits associated with better scores (conversions, trust, reduced bounce etc) are obviously still in play.



The tactics

Let's run through some common CWV culprits, with actionable tactics to improve LCP, FID, and CLS on your own site.

Improving LCP

- **Make sure [critical-path CSS for content](#) above the fold lives in <head>**

LCP measures the time to render the largest content element above the fold. To speed it up, place all style rules related to above-the-fold content into the <head> section of the HTML document. You can also inline the rules to eliminate the HTTP request(s) for the CSS file(s).

- **Make sure any [custom or heavy widgets](#) that require external API calls or third-party code (e.g. ads or social buttons) sit below the fold**

As LCP focuses on above-the-fold content, move everything that requires additional HTTP requests down on the page. This is especially important when connecting to third-party servers where you can't optimize server performance and response time.

- **Then, [preload anything above the fold](#), and defer loads below the fold**

Use the [rel=preload](#) attribute for <link> tags that load assets for above-the-fold content so the browser can request them first. Add the [defer](#) attribute to scripts that load below-the-fold content to delay their execution.

- **Be discerning about your pop-ups and plugins – does the value they deliver make up for the lags they introduce?**

Measure the load time of all non-essential elements to see how they impact your metrics – especially if their content appears above the fold. Use Real User Monitoring (RUM) to see the user impact of plugins, pop-ups, and other elements that require more HTTP requests or add extra weight. .

- **Reduce server response times by [caching assets](#)**

Cache everything you can to make your page load faster, including HTML pages, CSS stylesheets, JavaScript files, images, and other multimedia content. Combine different caching techniques such as site, browser, and server caching.

- **Optimize and [compress image](#) and text files**

Reduce the sizes of image and text files as much as possible. Minify, compress, and bundle them together using [HTML and CSS minification](#) tools, module bundlers, and image optimization tools.

- **Install a CDN to get around region-based navigation delays**

A [content delivery network](#) is a group of servers and data centers geographically distributed around the world. This means you serve static files from a server physically close to the user's location to improve load times and LCP scores.

- **Review your image sizing and [formats](#)**

Most browsers now support many image formats, including vector graphics and [next-gen images](#) such as AVIF and WebP. When choosing an image format, consider the technology behind it (e.g. lossy vs lossless compression), its weight, and browser support. Load responsive images for different viewport sizes using HTML features such as the [srcset and sizes attributes](#).

Improving FID

- **Compress and minify code in your web pages and script files**

Reduce the size of your dynamic content or scripts. While minification removes all whitespace and unnecessary code, compression modifies text data with a compression algorithm like the lossless [DEFLATE algorithm](#) used by Gzip.

- **Remove any unused CSS or JavaScript**

Use the [Coverage tab](#) in Chrome DevTools' Sources view to detect unused CSS and JavaScript on your page. Analyze and remove the unused parts. To save time, you can use tools like [PurgeCSS](#) or cleanup plugins for content management systems.

- **If you're using long-running code, break it into smaller, asynchronous tasks, and introduce lazy-loading**

[Code splitting](#) is a performance optimization technique that divides longer code into smaller bundles. You can load each bundle on demand and, if needed, in parallel with other bundles. Split your scripts either manually or by using module bundlers such as [Webpack](#), [Rollup](#), or [Parcel](#).

- **Load heavy scripts for less essential components off the critical path**

Remove scripts from the critical rendering path by moving them from the <head> section to the closing

</body> tag. If you need a script earlier in the page rendering lifecycle, keep it in <head> but use either the [defer or the async attribute](#). Both async and deferred scripts download in the background as non-render-blocking resources, but while 'defer' follows the document order, 'async' doesn't (the script loads independently from the rest of the page).

Improving CLS

- **Avoid having multiple heavy widgets on the same page**

Widgets are often slow to load and tend to render at different paces, resulting in jittery, disorientating layout shifts. Review and remove unnecessary widgets and explore lighter alternatives for important but heavy widgets.

- **Restrict the number of custom fonts used, ideally to a maximum of two**

There are [two main strategies](#) here: FOUT (Flash of Unstyled Text) where the fallback font is shown until the custom font is rendered and FOIT (Flash of Invisible Text) where invisible text is shown while the custom font is loading. As both can result in unexpected layout shifts, it's best to limit the number of custom fonts on each page. You can also [preload custom fonts](#).

- **Always specify dimensions on your images and video elements**

Add width and height attributes to all your image and video tags to make the browser allocate the right amount of space for these elements before their content is rendered. Define the height and width values in pixels.

- **Avoid dynamic ads and banners that will displace other page elements, and instead leave a large “placeholder” space for the ad to load into**

Similar to images and videos, always allocate sufficient space for ads and banners that load dynamically to prevent unexpected shifts. [Define a container](#) for each dynamic element and set dimensions based on viewport size.

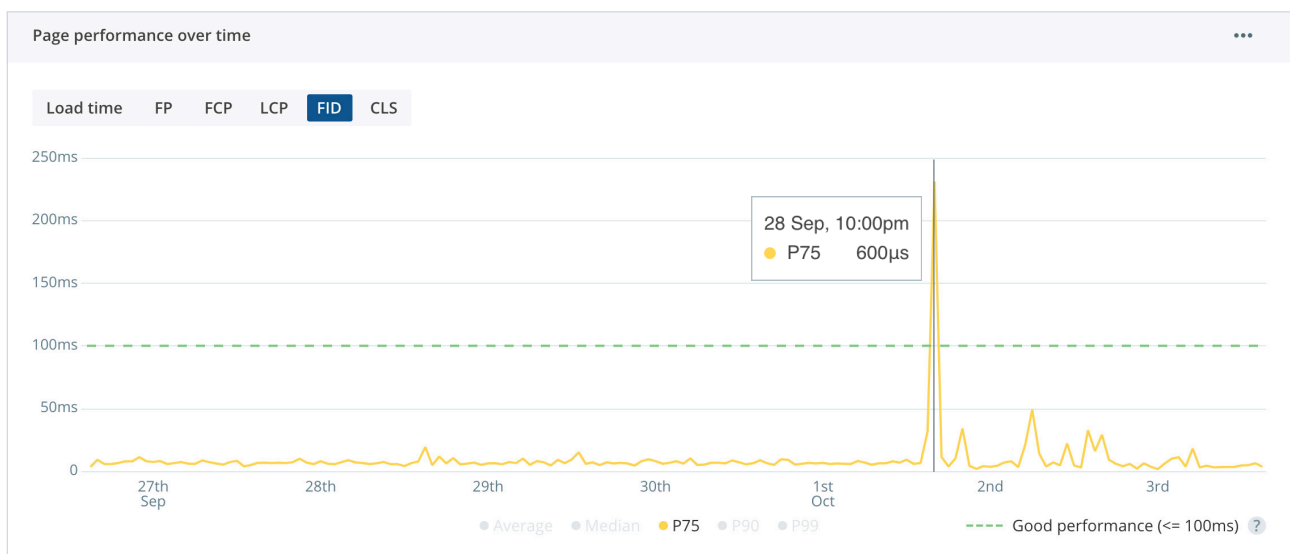
The workflow

Now that we've armed you with a comprehensive list of tactics, it's time to get to work. Implement our simple four-step workflow to identify opportunities, improve your Core Web Vitals, and delight your users.

To follow this workflow effectively, you'll need to equip yourself with a dedicated [Real User Monitoring tool](#), PageSpeed Insights, and Lighthouse.

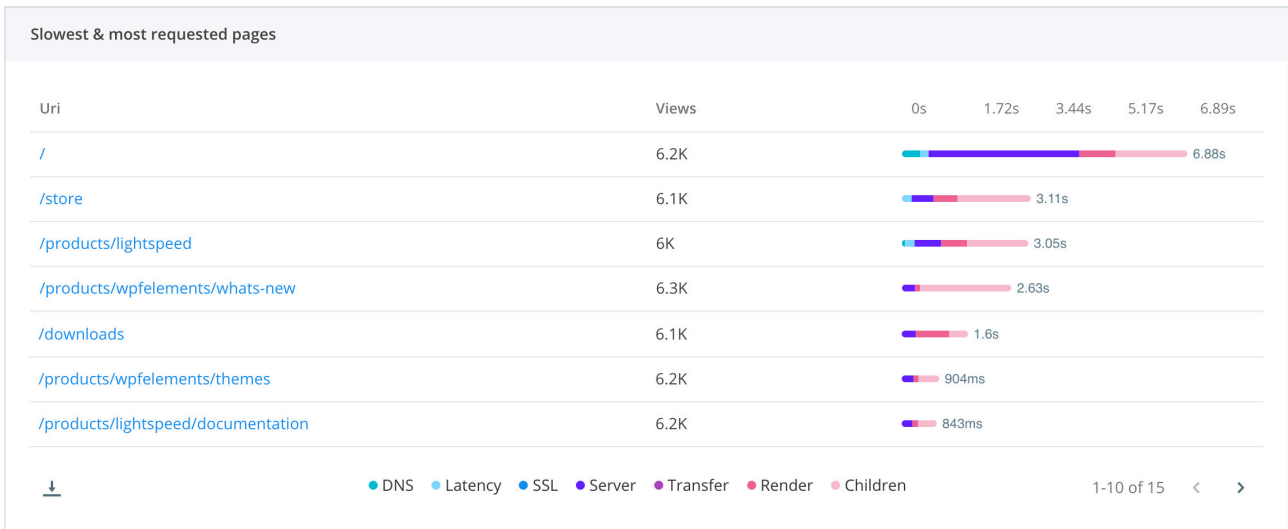
Step 1: Identify opportunities with RUM

First off, take a glance at how your website is performing across all three metrics with Real User Monitoring. In Raygun, you can use the P75 toggle to see how your site stacks up against each CWV at a high level, and detect if there are any recent spikes or trends that require immediate attention.



Since Core Web Vitals are calculated on a per-page basis, you'll now need to identify the pages of your site that offer the greatest opportunities for improvement. There are three things to consider when shortlisting pages for further discovery: traffic, importance of page, and room for improvement.

Modern RUM tools automatically group the slowest and most requested pages. This provides a great starting point to drill into individual pages and see how they score against each CWV. Follow this process to identify a single page, or even better a specific Core Web Vital on a page, and continue to step 2.



Step 2: Prioritize optimizations

Now that you know what you want to improve, it's time to figure out how. The fastest way to shortlist specific optimizations is by using PageSpeed Insights. Simply enter your page URL and PSI will automatically provide a list of actions, alongside their estimated impact.

At this point, you need to look into common CWV inhibitors and best practices specific to your tech stack and objective. There are lots of solid resources and developer blogs sharing first-hand experiences of improving Core Web Vitals – not to mention our curated list of tactics above.

For even greater context, you can validate ideas against real user insights. RUM gives you instance-level diagnostics on every page request with a full waterfall breakdown of load times across DNS, latency, SSL, server, and transfer times. Some RUM tools will even allow segmentation by geolocation, device and browser type.

Remember to take note of your CWV scores according to both CrUX and Lighthouse datasets, as these will be useful for further testing and iteration in step 3.

Show audits relevant to: All FCP LCP TBT CLS

Opportunities – These suggestions can help your page load faster. They don't **directly** affect the Performance score.

Opportunity	Estimated Savings
▲ Reduce unused JavaScript	2.62 s
▲ Serve images in next-gen formats	2.1 s
▲ Reduce unused CSS	1.19 s
■ Properly size images	0.6 s
▲ Reduce initial server response time	0.54 s

document short because all other requests depend on it. [Learn more](#). [FCP](#) [LCP](#)

URL	Time Spent
https://developers.google.com	640 ms

Step 3: Implement changes, learn, and iterate

Using the information gathered in step 2, it's finally time to take action on the CWV scores that are hindering your website performance. After making an optimization, refer to your Lighthouse scores. Lighthouse uses lab data, meaning you can see changes to your Core Web Vitals immediately after implementing optimizations, without having to wait for real users to interact with your page. This is an effective way to get an early indication of the impact of your changes.

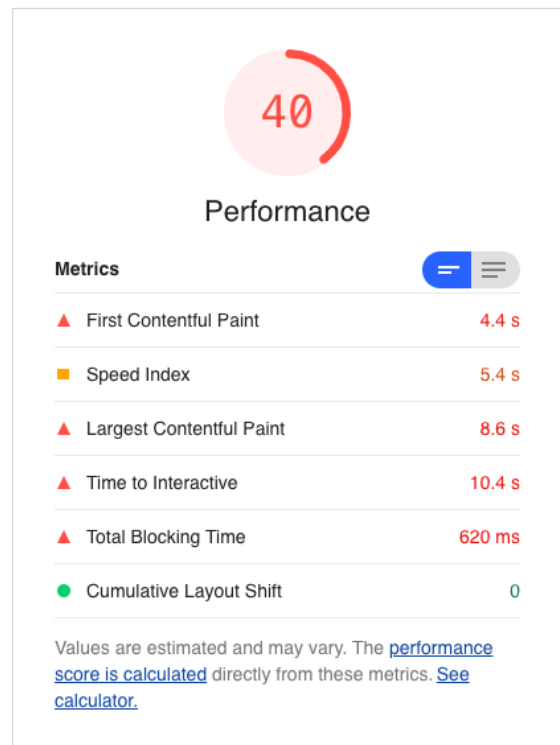
Next, you'll need to see how these changes manifest in the hands of your real users. While the lab data from Lighthouse was a good initial guide, you'll need to cross-check your improvements with a Real User Monitoring solution. RUM depends on real website traffic, so the amount of time you'll need to wait in order to get meaningful insights will vary from page-to-page, site-to-site.

With enough traffic, you can use RUM to see the impact of your changes soon after they go live, by adjusting the timeframe to only show new traffic since the last deployment. RUM will show the impact of optimizations on your real user's digital experience. Use this data to learn quickly and make informed decisions about what worked and what didn't.

Step 4: Monitor trends overtime

RUM lets you track the success of your changes over time, to keep a keen eye on how your CWV scores are trending. You can monitor and compare specific timeframes, or view the line graph to see trends of each individual metric over time. We highly recommend setting up alerts in RUM to notify you if any CWV metric drops below a standard that you're happy with. This will give you peace of mind that your users are consistently getting the experience they deserve.

28 days after implementing your improvements, check whether your Core Web Vitals have improved in Google's eyes (your CrUX data). This is the moment of truth and the final payoff, so we'll have our fingers crossed for you. You got this.

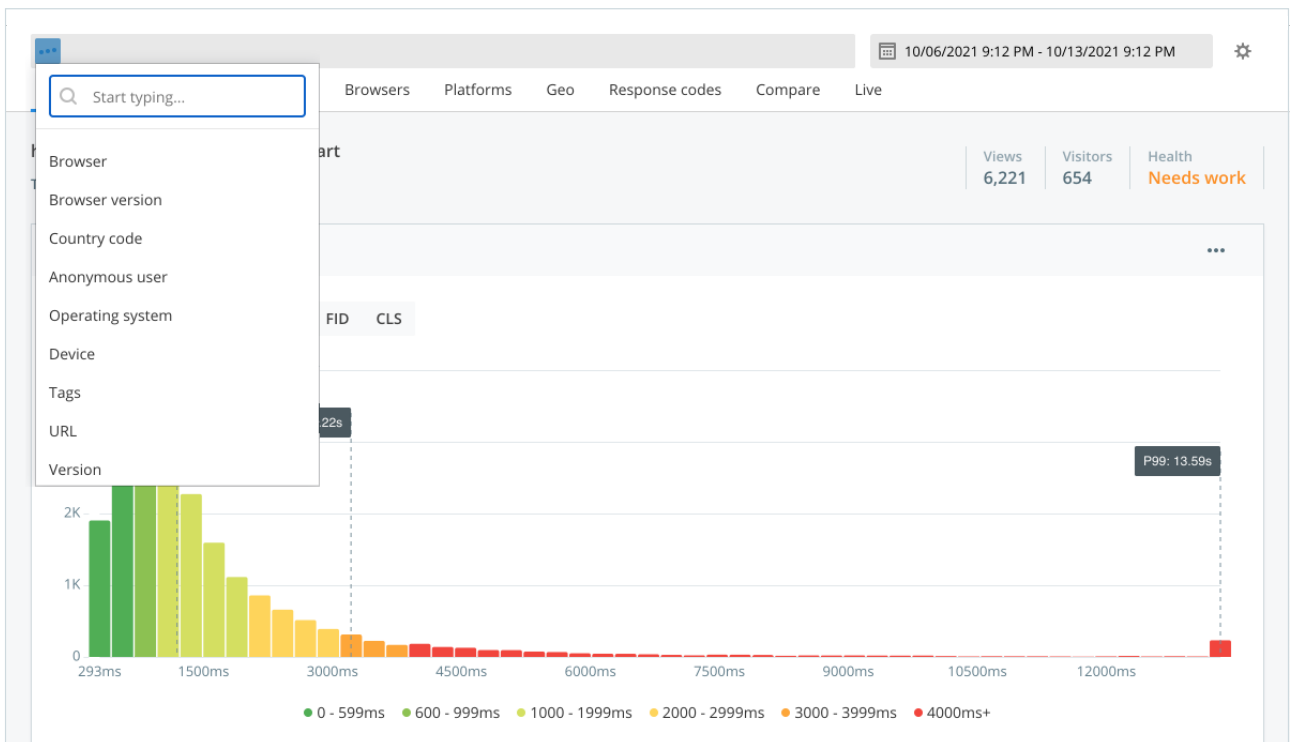


Core Web Vitals in Raygun

Let's take a look under the hood to understand what makes Raygun's first-class support for Core Web Vitals so powerful.

Filter data by what matters to you most

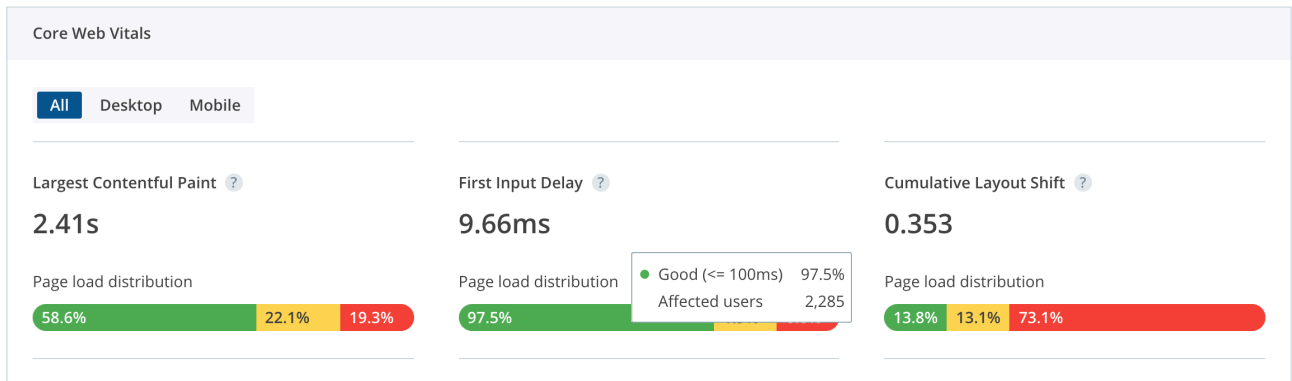
Raygun gives you complete control over user data so you can surface the insights that matter to you and your business. Overlay top-level filters such as country, browser, URL, OS, device, and even custom tags, to help narrow your focus.



Explore commonalities in users who have lower scores, or segment your audience to see how your most valuable customers are experiencing Core Web Vitals on your site. If the majority of your customers are based in the United States, for example, you can filter your traffic to see exactly how users in the U.S are experiencing each of your Core Web Vitals and optimize accordingly.

Unlock greater context in order to take action

Monitor how users are distributed across each Core Web Vital in the page performance histogram and see exactly who has been impacted by each Core Web Vital score. Spot problematic cohorts and hover and click into specific histogram bars or scores to see the number of users affected and the full list of users and sessions.

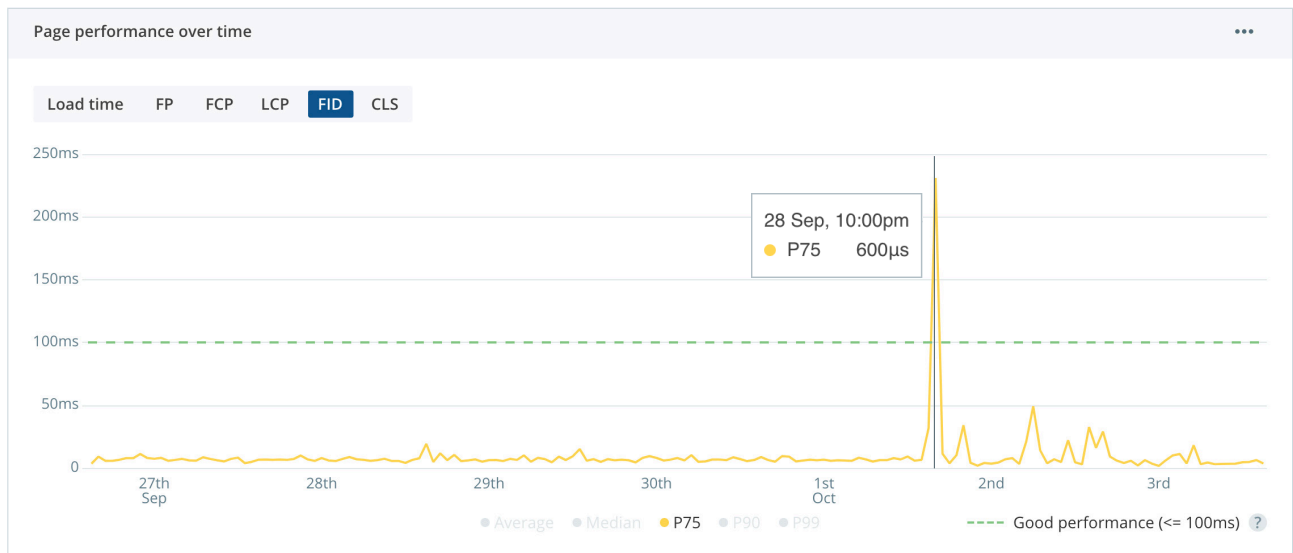


From here, you'll be able to see who makes up each of these cohorts, with contextual information such as location, browser, and operating system. Dive into individual sessions to see what pages they have visited along with performance diagnostics of every page request showing a full waterfall breakdown of load times across DNS, latency, SSL, server, and transfer times. This will give you the power to easily spot issues and replicate them on your own device.

Session ID	Views	Browser	OS	Location
#598157485052	1	Mobile Safari	iOS	United States of America
#598157488601	1	Firefox Mobile	Android	United States of America
#598157490445	1	Firefox	Mac OS X	United States of America
#598157493897	1	Firefox	Windows 10	France
#598157508263	1	Safari	Mac OS X	Germany
#598157508266	2	Edge	Windows 10	United States of America
#598157511993	2	Edge	Windows 10	France
#598157513826	2	Mobile Safari	Android	Canada
#598157519127	1	Opera	Windows 10	United Kingdom
#598157520887	1	Chrome	Mac OS X	United States of America
#598157527780	2	Safari	Mac OS X	United Kingdom

View trends over time and set up custom alerts

Monitor how your Core Web Vitals are trending over time to track the success of optimizations on real users. By selecting the P75 filter on the performance over time graph, you'll be able to see where you fall on the spectrum of "Good", "Needs Work" or "Poor" for each of your Core Web Vitals.



Trends can be extremely valuable for identifying abnormalities in your scores over time, especially when used in conjunction with top-level filters. Monitor trends at an individual page level for greater visibility and context into what actions have the biggest impact on your page's performance so you can learn and optimize quickly to improve your scores.

By setting up custom alerts, you can ensure that you'll never miss a fluctuation in your CWV scores. When used in addition to top-level filters such as location, browser, URL, and more, you have complete control over what triggers an alert.



"Any customer UX issues, like poor performing pages, are escalated and dealt with very quickly, even with a high volume of traffic coming to our website."

Clinton Van Ry
Peach



"I would absolutely recommend Raygun. It's easy to get started and you'll quickly get real visibility into what your users are actually experiencing."

Greg Shackles
Olo

Taking control

Google's introduction of Core Web Vitals gives you a chance to boost customer experience and prove the value of your work. You need to act fast to gain competitive advantage and capitalize on these measures – CWV, and the broader Web Vitals program, will continue to evolve and expand as Google builds and iterates on them as a measure of experience. But while the latest metric may change, search rankings and business success will only be increasingly tied to the perceptions of the user.

While you can make crucial progress by addressing CWV now, creating an overall culture of customer-centric development will keep you ahead of the curve going forward. Investing in Real User Monitoring as a critical connection between your code and your UX will set you up for ongoing success.

Ready to boost your CWV, increase your ranking and get your entire business on board with performance? [Try Raygun Real User Monitoring](#) today.

[Try Raygun in AWS Marketplace](#)



What is AWS Marketplace?

AWS Marketplace is a curated digital catalog that makes it easy for organizations to discover, procure, entitle, provision, and govern third-party software. You can find thousands of software listings from popular categories like security, business applications, and data & analytics, and across specific industries, such as healthcare, financial services, and public sector. You can also explore and buy professional services to configure, deploy, and manage your third-party software.

With AWS Marketplace, you can shorten procurement times, implement the controls you need to operate with confidence, and enable your organization to unlock innovation. [Find out more](#)

