A NEW APPROACH IN QUANTIFYING USER EXPERIENCE IN WEB-ORIENTED APPLICATIONS

Vasilije Vasilijević[1]
Nenad Kojić[2]
Natalija Vugdelija[3]

DOI: https://doi.org/10.31410/ITEMA.2020.9

Abstract: *The primary goal of every developer is to develop the highest quality web application. The quality of the application is not only a subjective assessment of the developer, but objective and representative criteria for measuring performance must be defined. Google provides a model called Web Vitals with a subset of core Web Vitals that are important for quantifying user experience on the web. Some of the metrics are LCP (Largest Contentful Paint, refers to loading), FID (First Input Delay, refers to interactivity) and CLS (Cumulative Layout Shift, refers to visual stability). This paper will present modern technologies and tools for measuring the performance of websites and analyze them on a real example of a web application. The analysis will include the use and measurement of the most important parameters: Lighthouse, PageSpeed Insights, Chrome DevTools, Search Console, web.dev's measure tool, the Web Vitals Chrome extension and Chrome UX Report API.*

Keywords*: Web sites, Measuring, Performance, Web vitals.*

INTRODUCTION

Intensive development of the IT industry is a consequence of a large number of consumers of IT services. Such a drastic growth of IT services is primarily based on the improvement of network infrastructure, lower prices of hardware and Internet services. The large number of users who use an increasing number of services has led to the need to improve the entire network infrastructure and enable quality and fast transmission of signals and services (Velinov, 2020). This infrastructure is changing the way the Internet is used and now companies are primarily using websites as a way to present their services and products, and users to find them. A new concept of digital marketing is being created (Star, 2019) that suits new consumers and new ways of doing business. A large number of websites leads to the need for users to opt for some of the offered ones. The user's decision is based on their user experience and the experience that the site leaves them.

The information found on websites should be easily searchable, clear and accessible on different types of devices (Rosenfeld, 2015). Thus, the quality of content search has become extremely important (Drutsa, 2019) because it finds the user what he wants to get in return. The quality of the search is affected by the quality of the website and a large number of

[1] Academy of Technical and Art Applied Studies Belgrade - Department School of Applied Studies for Information and Communication Technologies, Zdravka Celara 16, Serbia
[2] Academy of Technical and Art Applied Studies Belgrade - Department School of Applied Studies for Information and Communication Technologies, Zdravka Celara 16, Serbia
[3] Academy of Technical and Art Applied Studies Belgrade - Department School of Applied Studies for Information and Communication Technologies, Zdravka Celara 16, Serbia

techniques and activities are needed for the website to be SEO optimized, and thus have a better position in web search engines (Patil, 2020). In addition to the technical parameters that indicate the quality of the website, one of the key roles is the impression of the user after his arrival on the site. The quality of the user experience is key to the long-term success of any website. The quality of the user experience is based on performance that can be classified and analyzed in a number of ways. A large number of authors have contributed to the development of various tools and techniques for measuring user experience (Castaneda, 2007) and this process is constantly evolving.

For this reason, Google has created a model for providing development guidelines to developers in the process of designing and building web applications, which are extremely important for providing an adequate user experience on the web. This model is called Web Vitals (www.web.dev/vitals). The indicators defined in this way are an empirical measure of the actual user experience that Google analyzes and which are based on the key needs of users: page loading speed, application interactivity, stability, content experience, etc.

In this paper, a set of specific factors called Core Web Vitals will be presented, which represent a general picture of the experience of web pages from the user's perspective, as well as techniques and tools for measuring them. The selected factors will be explained and implemented in a real web application in order to indicate the way in which it is possible to achieve optimal performance of a website.

The work is organized through four chapters: After the Introduction, a selected set of Core Web Vitals factors will be presented. The chapter results will show the quality of their implementation on a real website and at the end will give a conclusion and further guidelines in the work.

CORE VEB VITALS

As already mentioned, Core Web Vitals are a subset of Web Vitals that are used to measure specific aspects of the user experience when visiting a website. These parameters can be measured and displayed in a large number of Google tools. In this paper, some of them that the authors consider the most important and which in their practice give the most important results of the user experience will be performed.

LCP (Largest Contentful Paint)) is the largest content color related to reading (loading) and measurement time when the most effective content of the page is probably displayed. The satisfaction of the range in which the largest page contents should be displayed is from 0 to 2.5 seconds from the first start of loading the pages.

FID (First Input Delay) represents time needed for web page to respond to user action. It means, when page is loading, and user tries to communicate with page through some action (e.g. button click), browser starts to count time needed to handle user request, and that time is called FID. If possible, that time should be less than 100 milliseconds.

CLS (Cumulative Layout Shift) is a cumulative layout shift that represents quantity of shift related to elements in a viewport of a web page. If we want, the user not to experience a bad experience in terms of content stability, the CLS should be less than 0.1.

L (Lighthouse) is a tool built in Chrome DevTools that serves for detection of problems that web developers are facing when it's about Core Web Vitals, Progressive Web Apps, Best Practices, Accessibility and Search Engine Optimization in development environment. It provides possibility to generate report based on these categories to get a deeper explanation of every problem, with tips on how to improve every metric. Report can be generated for both mobile and desktop results. In order for the measurements to be credible, it is necessary to open the Lighthouse in Incognito mode.

LCI (Lighthouse CI) allows measurements on pull requests before merge and deployment are done.

SC (Search Console) is a platform that provides traffic data from Google and helps developers to detect pages on their website that needs to be improved, based on that data. When SC generates report for every page, then PSI (Page Speed Insights) can be used to show opportunities to improve poor performance pages.

PSI (**PageSpeed Insights)** tool allows developers to enter page URL in order to get results of Core Web Vitals metrics. It is based on real traffic data and perspective of users that are visiting page, and also on lab data because Lighthouse is incorporated in generating report, so report looks similarly, with opportunities for optimization and steps that were passed through generating report. Performance result is expressed in percentiles. Best performance result is between 90-100 percentiles. PSI is good when there is a need to share results of metrics with someone, and also when only performance results are needed.

CUXR (**Chrome UX report** uses Core Web Vitals data of real users on your or competitors' website. So, main purpose is to help developers to realize how users are experiencing pages by tracking metrics of loading, interactivity and stability. That tracking is called RUM (Real User Monitoring). Developers enter URL and get raw API results from previous 28 days. If they have a pretty basic knowledge of SQL, they can query that results based on different parameters on Google BigQuery database. Difference between CUXR and PSI and Lighthouse is that CUXR does not provide list of issues that you can fix. With CUXR, if there is a need for data visualization, using CUXR Dashboard it's very easy to represent results visually with charts. Dashboard is provided by Data Studio and it connects result data on BigQuery, eliminating needs to create charts manually.

CDT (Chrome DevTools) are tools that mostly every developer uses in development process. In Performance panel, now layout shift can be identified in an Experience section. By selecting Layout Shift there are further information about shifts detected, and if there is a need to see which element caused shift, get mouse over "Moved from" and "Moved to" fields. Also, on the bottom of the Performance panel, there is an information about TBT (Total Blocking Time) which is related to FID.

WDMT (web.dev Measure Tool) is also a tool that uses Lighthouse to generate report about Core Web Vitals. The procedure is the same, provide URL of your page, and get metrics and tips that tell you about fixing issues.

WVCE (Web Vitals Chrome Extension) is available in Chrome Web store. It can be added to Chrome and used for measuring vitals on website including LCP, FID and CLS. When it's installed, user need to navigate to URL that wants to measure, and after that, badge icon

provided by extension will update to red or green, depending on metric result. Clicking the icon, user can see further details about every metric.

WVJSL (Web Vitals JavaScript Library) is a way of measuring Web Vitals using JavaScript API. Library is imported into project with three functions for getting LCP, FID and CLS. To that functions, developer can pass custom function to send results to database or to do whatever he wants. This way of measuring matches with all above mentioned tools.

I0 (Image Optimization) The main problem with poor performance is always related to the images that are used and which are the most difficult type of resource. Image optimization started by reducing the weight of the images. JPEG 2000 progressive and WebP format are used for that. With them, incredible savings in image weight is achieved. You should be careful with the WebP format because it is not supported in all browsers, so use the JPEG format just in case. Images that are important for uploading are uploaded to the CDN. It is recommended that all images be on the CDN as well but in this example the other images are hosted locally. Depending on the device, media queries for images of different sizes are used. Lazy loading images technique is used to defer images outside the viewport. Using lazy sizes, the off-screen image storage is accomplished. It is also possible to defer images loaded via CSS using lazy sizes bgset.

FM (File minification) CSS and JS were minified. Gulp is used to minify these files. It is also possible to do this via online tools or WebPack using e.g. css-mini-extract-plugin for css minification. Unused CSS is removed using the purgecss tool. It is necessary to give the path to the desired CSS file as a parameter, and the output is cleared CSS from parts that are not used. The important thing is if you use Bootstrap to clean the file from classes that you do not use. Unused JS and CSS can be found in Google Chrome DevTools, the Sources tab in the Coverage tab. You can start recording and interact with the whole page, and finally see which parts of the code are not used, and delete those parts.

GC (Gzip compression) Compression was performed on the server. In addition to gzip, you can also use Brotli.

TPR (Third-party resources) For the resources, attributes rel = "preconnect", rel = "preload" or rel = "dns-prefetch" are added.

RBR (Resources that block rendering) CSS resources that block rendering are removed.

CRP (Critical render path) A critical render path needs to be done. Critical CSS is detected using a tool called PentHouse. It is best to put critical CSS in the style tag in the head section, to get improved FCP improved.

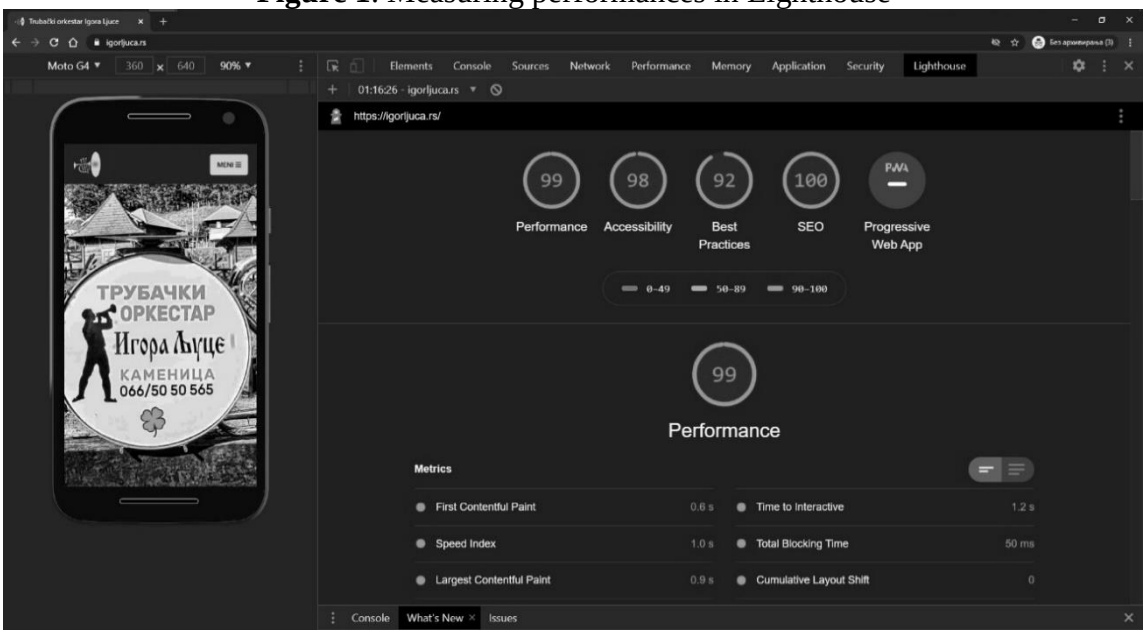
C (Caching) Asset caching has been done, which does not change often on the server.

DSBR (Delay scripts that block rendering)
Resources that block rendering can be deferred as needed to avoid blocking the main thread. It is also necessary to add async if we want the script to be downloaded asynchronously.

RESULTS`

Use case 1 - Example of measuring performances in Lighthouse. When we want to use Lighthouse, we should go to webpage that we want to test, in Incognito mode, and with Right Click go to Inspect Elements. Now, we should navigate to the Lighthouse Tab in DevTools menu on the top.

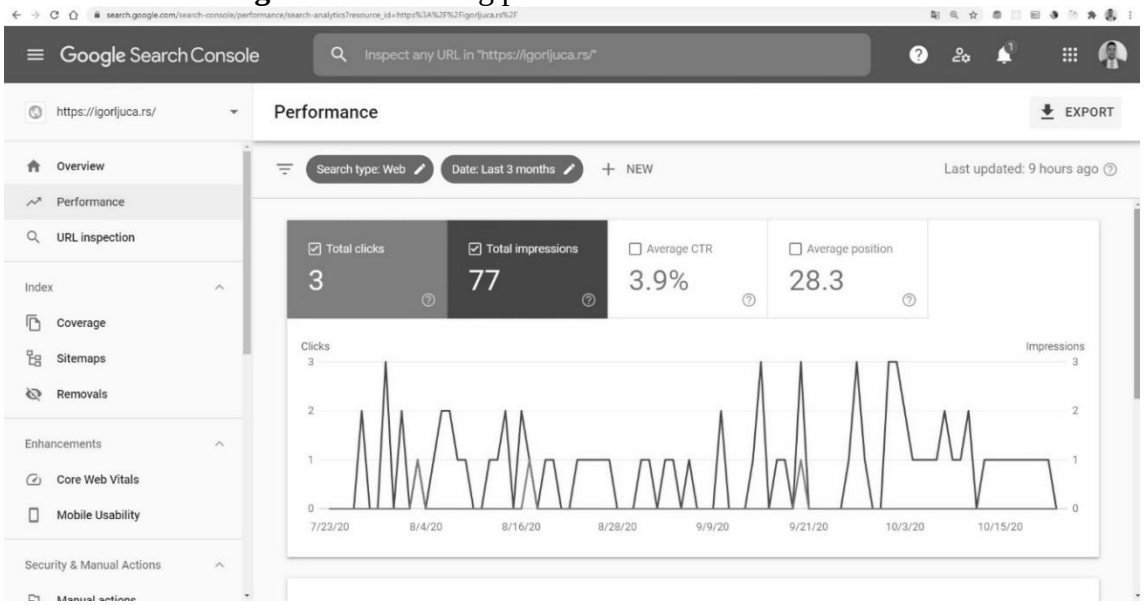
Figure 1. Measuring performances in Lighthouse



When we got to the Lighthouse, we can choose categories that we want to measure. For our case, we will select Performance category, then click to Generate Report. After that, we are getting results of measuring as it is displayed on the Figure 1.

Use case 2 - Search Console is used to identify pages that need to be more optimized for users. Also, we can see much more information about our website, e.g. performance, mobile usability, sitemaps, coverage, removals... This is shown on Figure 2.

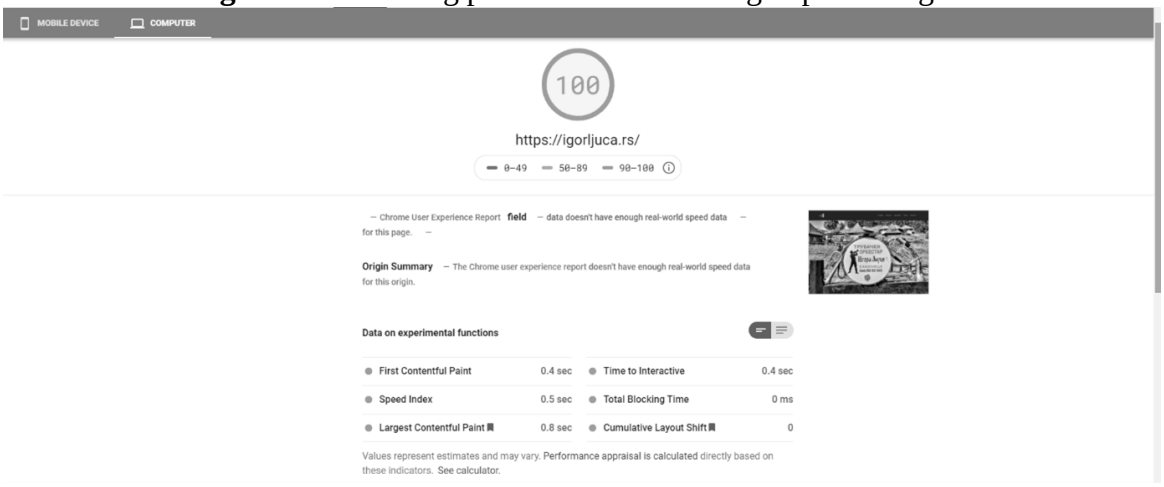
Figure 2. Measuring performances in Search Console



Search Console Core Web Vitals section use CUXR to get data about user experience on our page.

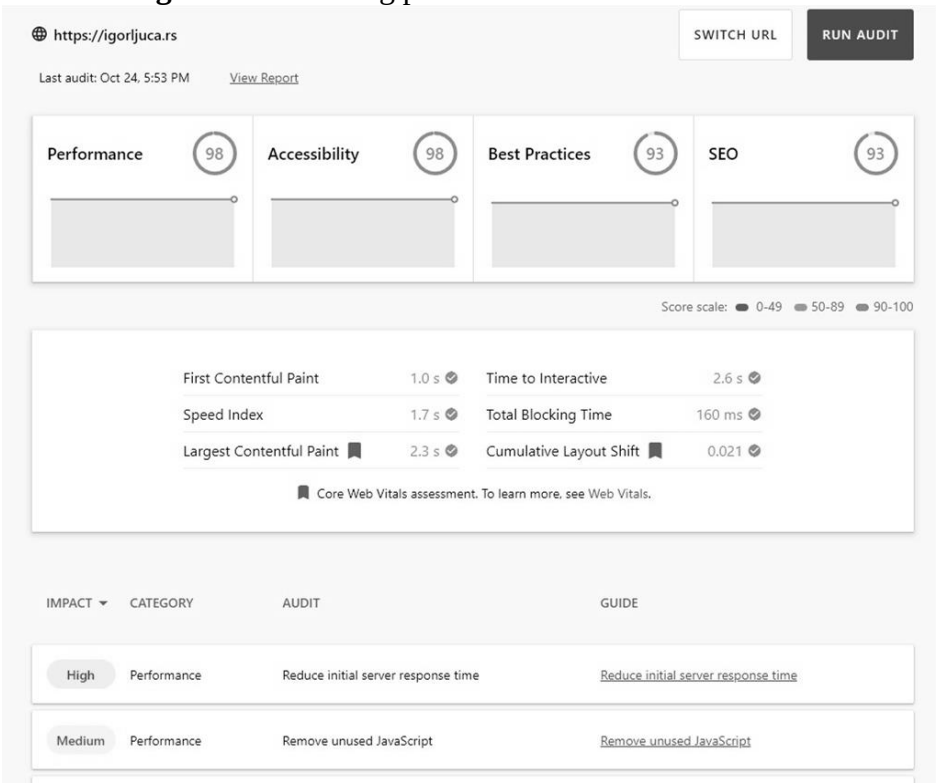
Use case 3 – Another example of performance measuring is with PageSpeed Insights. Unlike Lighthouse, here we have real data based on our user experiences. Only thing that we have to do is to provide URL of your webpage and to type it in input field. Then click Analyze and see metrics that Google collects from our users, with opportunities how to more optimize our performance and information about passed audits as shown in Figure 3.

Figure 3. Measuring performances with Page Speed Insights



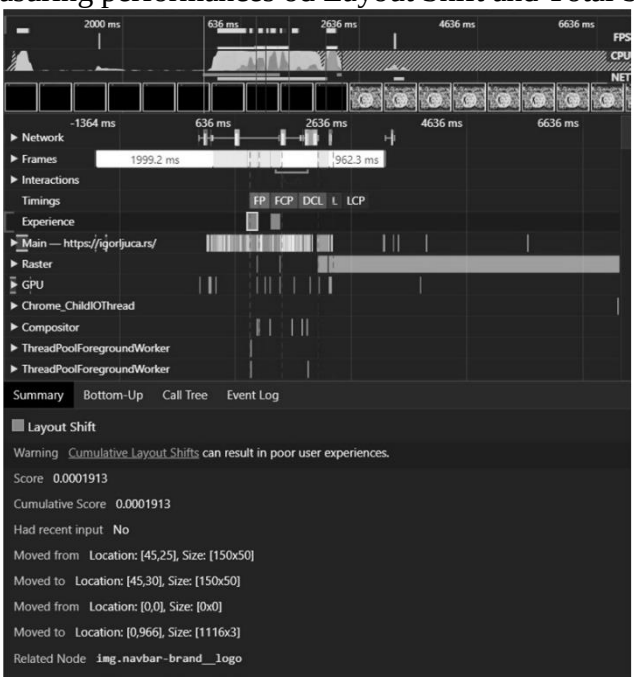
Use case 4 – If we want to measure Web Vitals on Google web.dev site we can simply go to their website web.dev/measure and type URL that we want to inspect. Displayed result on Figure 4 are similar to Lighthouse and PageSpeedInsights.

Figure 4. Measuring performances with web.dev site



Use case 5 – This is example of Chrome DevTools Performance Panel where we can detect Layout Shift and Total Blocking Time on our page, Figure 5.

Figure 5. Measuring performances od Layout Shift and Total blocking Time



If we go to DevTools, then go to Performance Tab we should click Record. After that, refresh our page and stop recording. We will see Layout Shift if exist and in Summary tab we can get more information about shift Location. Total Blocking Time is displayed in footer of Performance Tab.

All these examples show a part of the possibility of advanced application of measuring the user experience of web applications, and in addition to the above, many others can improve the quality of the final evaluation of the website.

CONCLUSION

This paper presents a set of factors that are an integral part of Core Web Vitals. These factors represent an empirical indicator of user experience when evaluating websites and are available in a number of Google tools. As there are a large number of factors that affect the overall user experience, this paper singles out only some that the authors considered to be the most important and given explanations, methods of use and results of their application in a test example of the website.

In general, the results that can be measured are primarily the quality of work of web developers who participated in the creation of the website. These indicators are highly correlated with the attention that developers analyze in the user experience. A large number of applications do not have enviable results and these tools can contribute a lot to paying more attention to these parameters when they can be easily measured and analyzed.

Further work will be focused on the implementation of additional factors and improving the user experience of web applications.

REFERENCES

Castaneda J. A. & Muñoz-Leiva F.& Luque T. (2007). Web Acceptance Model (WAM): Moderating effects of user experience, Journal of Information & Management Vol. 44, Is. 4, (pp. 384-396), Elsevier.

Drutsa, A., Gusev, G., Kharitonov, E., Kulemyakin, D., Serdyukov, P., & Yashkov, I. (2019, July). Effective Online Evaluation for Web Search. In Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval (pp. 1399-1400).

Karthikeyan, R., Michael, G., & Kumaravel, A. (2017). A HOUSING SELECTION METHOD FOR DESIGN, IMPLEMENTATION & EVALUATION FOR WEB BASED RECOMMENDED SYSTEMS. International Journal of Pure and Applied Mathematics, 116(8), 23-28.

Patil, S. A. (2020). Comparative SEO Techniques Analysis on core WebPages and its Effectiveness in Context of Google Search Engine, International Journal of Scientific Development and Research, Vol 5, Is.3 (pp. 420-428)

Rosenfeld, L. & Morville, P. & Arango, J. (2015). Information Architecture: For the Web and Beyond, O'Reilly Media.

Star, D. (2019). Digital Marketing 2020: Grow Your Business with Digital Marketing, Kindle Edition

Velinov, V. (2020). MODERN NETWORK ARCHITECTURES NETWORK AND SYSTEM PROTECTION, Kindle Edition

www.web.dev