Development Technologies Impact in Web Accessibility

Carlos Duarte[1], Inês Matos[2], João Vicente[2], Ana Salvado[2], Carlos M. Duarte[2], Luís Carriço[1]
LaSIGE, Faculdade de Ciências, Universidade de Lisboa
Campo Grande
Lisboa, Portugal
[1]{caduarte, lmcarrico}@fc.ul.pt [2]{imatos, asalvado, jvicente, cduarte]@lasige.di.fc.ul.pt

ABSTRACT
This paper presents a study assessing the effect of development technology choices on the accessibility quality of web pages. We used an off-the-shelf tool to identify technologies over a set of 1669 pages, each from a different domain. We have focused, in particular, in programming languages, Web and JavaScript frameworks, and content management systems categories. Simultaneously, we used an automatic accessibility evaluation tool to assess the accessibility of those pages. The study shows statistical evidence that some technologies lead to more accessible pages, within all categories.

CCS Concepts
• Accessibility →Empirical studies in accessibility • Human-centered computing → Empirical studies in HCI • Human computer interaction (HCI) → Web-based interaction.

Keywords
Web Accessibility; Automatic Evaluation; Web Technologies.

1. INTRODUCTION
Users with disabilities face significant challenges while browsing the Web. Although research efforts [1], guidelines [2] and public policies have tried to ameliorate the situation [4], change in the Web tends to outpace accessibility remedies [9].

A Web page is no longer a static HTML document delivered by a server. Instead, it is a browser's interpretation of both content and logic, delivered by complex client and server programming languages. They often reuse several toolkits and frameworks, and are created through intricate development tools [3]. The amount of available development techniques is enormous, which makes it even harder to accompany the development pace [9].

Some providers of development techniques commit to include accessibility guidelines conformance in their solutions. Yet, the experience with accessibility logos [11] shows that stating it and actually doing it is a very different thing. Moreover, providing support for accessibility conformance does not mean that developers actually take advantage from it. Again, even if by law pages should be accessible in many countries, the fact remains that they are not [5].

It is therefore of importance to understand how commonly used development technologies impact on the accessibility quality of the produced webpages. In this way, responsible developers can be advised on what to use and what to avoid. Moreover, further pressure is made on providers to improve their development techniques. Eventually, if everybody is engaged, either by law, by force, by education or by moral responsibility, the web becomes a more accessible place.

This paper presents a preliminary study addressing the impact of some of those technologies in the web accessibility. We randomly selected 1669 pages of different domains and identified the technologies, from programming languages to content management systems, which were used in their development. Then, using QualWeb [7], an automatic accessibility evaluation tool, we assessed the quality of those pages. The results demonstrate that it is possible to identify technologies in each category that actually lead to more accessible pages. It is also statistically shown that some technology providers committed to accessibility, actually offer solutions that drive developers to produce more accessible pages - although some better than others. Interestingly, even at the server programming language level, the study reveals that its adoption impacts significantly in the accessibility outcomes.

2. METHODOLOGY
For the accessibility comparison study, we began by crawling the web using crawler4j (http://github/yasserg/crawler4j). We used the following crawling seeds: www.alexa.com, www.wikipedia.com, moz.com/top500, www.sapo.pt and pplware.sapo.pt. The websites were selected based on their high number of outgoing links to many different topics. For the study we considered the first page crawled of the first distinct 1669 domains that were visited.

The crawled webpages were then subject to two analyses. Their accessibility was automatically evaluated using QualWeb [7]. QualWeb is an automatic accessibility evaluation tool that is capable of evaluating pages after all browsing processing, just before rendering. The advantage is that the evaluation is done on the pages that are actually delivered to the user, which in fact is the target of the WCAG. Previous studies showed significant accessibility differences between pages as delivered by the server and those evaluated after browsing processing [6]. It adopts WCAG 2.0 and implements 44 HTML & CSS techniques. QualWeb is available through a web interface, a browser plug-in and a script-based application. The latter was used.

The second analysis was conducted with Wappalyzer. This is an open source tool (http://wappalyzer.com) that can be used to identify the technology applied in the creation of webpages. It is commonly available as a browser extension. However, given that the source code is available, we adapted it to be used as a script to be applied on the crawled webpages. Wappalyzer is capable of identifying technologies belonging to 51 different categories. For this study, we considered four of these categories: Programming languages, Web frameworks, JavaScript frameworks and Content management systems. We selected these categories since the technologies they represent have direct impact on the rendering of the pages, thus on their potential accessibility.

W4A'16, April 11 - 13, 2016, Montreal, Canada

ACM 978-1-4503-4138-7/16/04…$15.00
DOI: http://dx.doi.org/10.1145/2899475.2899498

In the data analysis, we considered only technologies that were used in, at least, 15 different pages. Table 1 presents those that meet this requirement, for each category, with the total number of webpages where they are used.

Table 1. Technologies considered in the accessibility evaluation

Category	Technology	# of pages
Programming Languages	Java	31
	Ruby	26
	PHP	450
Web Frameworks	ASP .NET	85
	Ruby on Rails	23
	Twitter Bootstrap	179
JavaScript Frameworks	jQuery	744
	Modernizr	151
	MooTools	16
	Prototype	17
	RequireJS	24
Content Management Systems	Drupal	26
	Joomla	16
	WordPress	194

For each webpage we considered the quality of accessibility computed by the three metrics proposed in Lopes et al. [10], based on the number of passes (P), fails (F) and warnings (W) reported by the automatic accessibility evaluation: Conservative, which considers warnings as fails (P/P+W+F); Optimistic, which considers warning as passes (P+W/P+W+F); and Strict, which dismisses warnings (P/P+F).

We performed two analyses of the data. In the first, we compared the accessibility scores of webpages using each technology, to the accessibility of all the 1669 webpages crawled. Given that the sample did not follow a normal distribution, we used the non-parametric Wilcoxon test. Considering that we performed 14 tests (one per technology), the target significance level was p=0,05/14 =0,0038.

In the second analysis, we studied the effect of technology in each of the four categories considered. Here, we only used the strict metric based on the assumption that it the most accurate, since it considers that the P/F ratio within warnings follows the one computed when warnings are ignored [4]. Once again, we used a non-parametric test: Kruskal-Wallis, with post-hoc comparisons when applicable. Given that Kruskal-Wallis demands the samples in each group to be independent, we removed from this analysis the webpages that used more than one technology from the same category (e.g. ASP .NET and Twitter Bootstrap). Overall only 9 webpages were removed, and no more than 3 per category.

3. RESULTS

3.1 Accessibility by Technology

Table 2 presents the scores of the strict accessibility metric and the Wilcoxon tests with adjusted significance levels, for all the assessed technologies. Data shows that the use of technologies like PHP, Bootstrap, jQuery, Modernizr, Drupal and Wordpress have a positive impact on the accessibility of the resulting pages. The accessibility of those pages is statistically better (p<0,001) than the average, according to the strict and conservative metrics. Significance still holds for Bootstrap under the optimistic metric.

Table 2. Wilcoxon tests for the Strict accessibility metric, with significance levels already adjusted. Significant results are shown underlined and bold.

Technology	N	Strict		
		Average	Z	p
All	1669	0,60162		
Java	31	0,5615	-0,784	0,433
Ruby	26	0,6671	2,400	0,016
PHP	450	**0,6303**	**5,775**	**<0,001**
ASP	85	0,5891	-0,370	0,711
Ruby on Rails	23	0,6609	1,977	0,048
Bootstrap	179	**0,6760**	**7,582**	**<0,001**
jQuery	744	**0,6243**	**6,042**	**<0,001**
Modernizr	151	**0,6452**	**4,527**	**<0,001**
MooTools	16	0,6293	1,190	0,234
Prototype	17	0,5853	-0,829	0,407
RequireJS	24	0,6517	1,629	0,103
Drupal	26	**0,6980**	**3,645**	**<0,001**
Joomla	16	0,6533	1,758	0,079
WordPress	194	**0,6362**	**5,206**	**<0,001**

3.2 Accessibility in each Category

3.2.1 Programming Languages

A Kruskal-Wallis test showed that there was a statistically significant difference in the Strict accessibility metric score between the three different programming languages considered, $\chi^2(2)=7,482$, $p=0,024$.

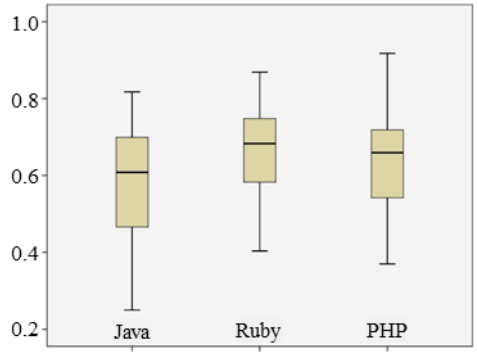

Figure 1. The impact of the three most common Programming languages of the sample in the webpage's accessibility.

Post-hoc comparisons found a statistically significant difference (p=0,024) between Java (M=0,5615, SD=0,167) and Ruby (M=0,6671, SD=0,1229). No differences were found between Java and PHP, or between Ruby and PHP (see Figure1).

3.2.2 Web Frameworks

The Kruskal-Wallis test found a statistically significant difference in the strict accessibility metric score between the three web frameworks considered in this analysis, $\chi^2(2)=35{,}325$, $p<0{,}001$.

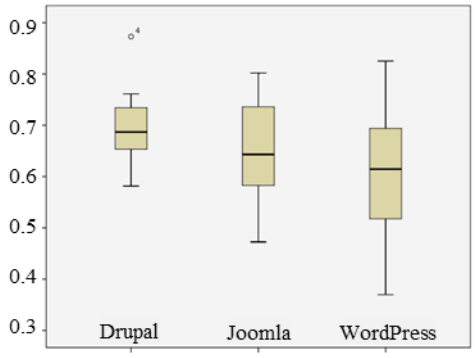

Figure 2. The impact of the three most common Web Frameworks of the sample in the webpage's accessibility.

Post-hoc comparisons found that ASP.NET (M=0,5891, SD=0,0899) had significantly lower accessibility scores than both Ruby on Rails (M=0,6609, SD=0,1288, $p=0{,}038$) and Twitter Bootstrap (M=0,676, SD=0,1001, $p<0{,}001$). No differences were found between Ruby on Rails and Twitter Bootstrap (Figure 2).

3.2.3 JavaScript Frameworks

No statistically significant differences between the strict accessibility scores of the five JavaScript frameworks considered was found by a Kruskal-Wallis test, $\chi^2(4)=7{,}986$, $p=0{,}092$.

3.2.4 Content Management Systems

A Kruskal-Wallis test showed a statistically significant difference of the strict accessibility metric score between the three CMS considered in this analysis, $\chi^2(2)=10{,}059$, $p=0{,}007$.

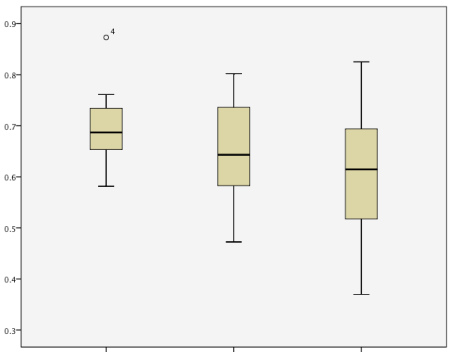

Figure 3. The impact of the three most common Web Frameworks of the sample in the webpage's accessibility.

Post-hoc comparisons found that Drupal (M=0,698, SD=0,1111) had significantly higher accessibility scores ($p=0{,}005$) than WordPress (M=0,6362, SD=0,1129). No differences were found between Joomla and the others (Figure 2).

4. DISCUSSION

Our first analysis of the data collected showed that some technologies have the potential to lead to webpages with higher accessibility scores than the average webpage. In particular, in our sample of 1669 webpages from different domains, we found that webpages using the following technologies had better accessibility scores, than the average: PHP, Twitter Bootstrap, jQuery, Modernizr, Drupal and WordPress. Bootstrap reveals an even stronger impact for better webpages as it keeps its statistical significance according to the optimistic metric.

It is also interesting to notice that although not statistically significant in this analysis, Ruby (M=0,6671, Z=2,400, $p<0{,}016$) and Ruby on Rails (M=0,6609, Z=1,977, $p<0{,}048$) also show potentially relevant results towards a better accessibility. On the opposite sense, Java (M=0,5615, Z=-0,784, $p<0{,}433$) and Prototype (M=0,5853, Z=-0,829, $p<0{,}407$) reveal a trend to negative results. That trend, even if not statistically supported, becomes significant for the Java case within the category analysis.

Overall, data shows that at least one representative from each technology category stands out in terms of accessibility. Albeit these are only preliminary results of an analysis that we wish to deepen, they strongly suggest that accessibility concerned developers have valid options to support their work, in categories with a direct impact on the way pages are built and rendered.

In the second analysis, we looked inside each category, to try to understand if some of the technologies have higher or lower potential to produce pages with increased accessibility. We found no evidence of differences between JavaScript frameworks. Nevertheless, two JavaScript frameworks stand out in the results of the first analysis, as the ones used in pages with high accessibility: jQuery and Modernizr.

In the other 3 categories, we did found differences. In what concerns web frameworks, the evidence points to lower accessibility of webpages that make use of Microsoft's ASP.NET technology, when compared to the other two alternatives evaluated: Ruby on Rails and Twitter Bootstrap. Cross-referencing with the first analysis, Bootstrap seems to lead to webpages with higher accessibility scores. This is, probably, a consequence of the inclusion of ARIA markup in many of Bootstrap's components (e.g. modal dialogues), which prompts developers to consider accessibility in their projects.

Regarding content management systems, the first analysis highlighted two – Drupal and WordPress – that produce webpages with higher accessibility scores. The second analysis showed that using Drupal leads to webpages with better accessibility than using WordPress. Searching plug-ins for both CMS, one can easily find several that are driven by the desire to increase the accessibility of the webpages generated by these systems. However, Drupal has made a commitment to ensure that all features of the Drupal core, from version 7 onwards, confirm with WCAG 2.0 and ATAG 2.0[1]. The results of this evaluation demonstrate the positive effect of this commitment.

Finally, the programming languages category presents us with results harder to reconcile. PHP was the only language that lead to webpages with increased accessibility, as identified in the first analysis. However, the p-value for Ruby in this analysis is also small (p=0,009), which hints at Ruby having also the potential to produce webpages with higher accessibility than the average.

[1] https://www.drupal.org/about/features/accessibility

Strengthening this finding, in the second analysis, we found that webpages built with Ruby tend to be more accessible than those build with Java. Contrary to the previous cases, there seems to be no specific support for accessibility in these languages, thus the only justification that comes to mind is that the programmers profile may be different, revealing different ethical concerns.

4.1 Limitations of the Experiment

Our experiment setting has some limitations that may preclude on the type of results that can be extrapolated, including:

- Data gathering: since we made two http requests for each webpage, one for the QualWeb evaluation and another for Wappalyzer classification, we cannot guarantee 100% that the webpages are exactly the same. However, considering that QualWeb's evaluation runs within its own thread (ensuring a minimal timespan between requests) and that we are assessing development technologies, the likelihood that the pages are radically different it terms of these technologies is residual;
- Automated evaluation: since this experiment is centred on automated evaluation, it shares all of its inherent pitfalls [12]. However, QualWeb does its evaluation in browsing post-processing which has not been addressed in available benchmarks, and has revealed quite good results even compared with user testing [8]. It is also clear that adopting user testing or expert evaluation in 1669 webpages would have not been an efficient choice.
- Techniques coverage: QualWeb does not fully cover WCAG. However, this does not invalidate that there are differences in accessibility that are imposed by development technologies, and that in the assessed subset, which is quite reasonable, some provide better results than others.

5. CONCLUSION AND FUTURE WORK

This paper presents a preliminary study of the impact of development technologies in the accessibility of the resulting webpages. We have shown that the right choice of technologies does have a significate impact in accessibility. We also showed that it occurs even at the programming language level, where Java seems to be the worst choice. Also, at the CMS and Web Frameworks level, the commitment made by the technology providers seems to have a positive impact in the resulting accessibility.

As for the future, we plan to consolidate the study in several directions. First, we aim to extend significantly the sample size. That will allow further sectorial analysis, extend the number of analysed categories and even address technology combinations.

Regarding the validity of the study, we also aim to extend the subsample of webpages evaluated by experts, thus mitigating the automatic evaluation syndrome. We will consider a solution that makes only one http request, although controlling subsequent request from dynamic pages will be more difficult to cope with.

Finally, we aim to dig further upon the causes of the impact of technologies in accessibility. A first step would be a per technique analysis followed by a thorough inspection of the rooted causes. A more complex one, mentioned above, would be to understand if the developers profile matches the technologies and influences the accessibility outcomes.

6. ACKNOWLEDGMENTS

This work was funded by FCT LaSIGE Strategic Project, ref. UID/CEC/00408/2013.

7. REFERENCES

[1] Brown, J; and Hollier, S. 2015. The challenges of Web accessibility: The technical and social aspects of a truly universal Web. ISSN 13960466. Retrieved from: http://journals.uic.edu/ojs/index.php/fm/article/view/6165. Date accessed: 15 Jan. 2016.

[2] Caldwell, B., Cooper, M., Reid, L.G., and Vanderheiden, G. Web Content Accessibility Guidelines (WCAG) 2.0. 2008. Retrieved from http://www.w3.org/TR/WCAG20/. Date accessed: 15 Jan. 2016.

[3] Casteleyn, S., Garrig'os, I., and Maz'on, J.-N. 2014. Ten Years of Rich Internet Applications: A Systematic Mapping Study, and Beyond. ACM Trans. Web 8, 3, Article 18 (July 2014), 46 pages. DOI=10.1145/2626369.

[4] European Commission 2012. Study on economic assessment for improving e-accessibility services and products. Final report. SMART 2009-0072, Technoside, Tech4i2, AbilityNet, and NOVA on eAccessibility impacts. Retrieved from http://www.eaccessibility-impacts.eu/. Date accessed: 10 Feb. 2016.

[5] European Commission 2013. Study on assessing and promoting e-accessibility. ISBN: 978-92-79-33183-1. DOI=10.2759/33027.

[6] Fernandes, N., Batista, A. S., Costa, D., Duarte, C., and Carriço, L. 2013. Three web accessibility evaluation perspectives for RIA. In Proceedings of the 10th International Cross-Disciplinary Conference on Web Accessibility (W4A '13). ACM, New York, NY, USA, Article 12. DOI= 10.1145/2461121.2461122.

[7] Fernandes, N., Costa, D., Neves, S., Duarte, C., and Carriço, L. 2012. Evaluating the accessibility of rich internet applications. In Proceedings of the 9th International Cross-Disciplinary Conference on Web Accessibility (W4A '12). ACM, New York, NY, USA, Article 13. DOI=10.1145/2207016.2207019.

[8] Fernandes, N., Guerreiro, T., Marques, D., and Carriço, L. 2015. Optimus web: selective delivery of desktop or mobile web pages. In Proceedings of the 12th Web for All Conference (W4A '15). ACM, New York, NY, USA, Article 20, 4 pages. DOI=10.1145/2745555.2746668.

[9] Harper, S. and Yesilada, Y. *Web Accessibility*. Springer, London, 2008.

[10] Lopes, R., and Carriço, L. 2010. Macroscopic characterisations of Web accessibility. New Review of Hypermedia and Multimedia, 16, 3. DOI=10.1080/13614568.2010.534185.

[11] Petrie, H., Badani, A., & Bhalla, A. (2005). Sex, lies and web accessibility: the use of accessibility logos and statements on e-commerce and financial websites. In Proceedings of Accessible Design'05. Swinton, UK: British Computer Society.

[12] Vigo, M., Brown, J., and Conway, V. 2013. Benchmarking web accessibility evaluation tools: Measuring the harm of sole reliance on automated tests. In Proceedings of the 10th International Cross-Disciplinary Conference on Web Accessibility (W4A '13). ACM, New York, NY, USA, Article 1, 10 pages. DOI=10.1145/2461121.2461124.